Adapting Tabu Search to Accommodate Online demand variations in a Data Network

BY

MOHAMMED MOINUDDIN RIZWAN FAROOQI

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER ENGINEERING

MAY 2010

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Mohammed Moinuddin Rizwan Farooqi** under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER ENGINEERING.**

THESIS COMMITTEE

Dr. Mohammed H. Sqalli (Chairman)

Dr. Sadiq M. Sait (Member)

Dr. Marwan Abu Amara (Member)

Department Chairman
Dr. Basem Al Madani

Dean of Graduate Studies
Dr. Mohammed Salam Zummo

7/8/10
Date

Dedicated to

My Beloved Parents and Sisters

ACKNOWLEDGEMENTS

All sincere praises and thanks are due to Allah (SWT), for His limitless blessings on us. May Allah bestow his peace and blessings upon his Prophet Mohammad (P.B.U.H.) and his family. Acknowledgements are due to King Fahd University of Petroleum & Minerals for providing the resources for this research.

I would like to express my profound gratitude and appreciation to my thesis committee chairman Dr. Mohammed Houssaini Sqalli for his indomitable support and advice. Without his sheer guidance and suggestions, this work would have become a daunting task. His extensive knowledge, patience, and experience made it possible to shape the thesis. My unrestrained recognition also goes to the committee members, Dr. Sadiq M. Sait and Dr. Marwan Abu Amara for their interest, invaluable cooperation and support. Their continuous support, advice and encouragement can never be forgotten. Also, I would like to express my deepest thanks to every instructor who contributed in building my knowledge and experience. Thanks are also due to faculty and staff members of the Computer Engineering Department for their cooperation.

I render my profound acknowledgements to my friends Taha Anwar and Syed Ilyas Mohiuddin for the countless hours we spent on discussing and proposing solutions for the problems that I faced. Acknowledgements are due to my friends and colleagues who made my stay at the university a cherished and an unforget-

table era. I would like to make a special mention for my friends Mansoor, Waseem, Naser, khalil, Minhaj, Khadir, Malik, and Abdul Kareem.

Finally, I also thank my beloved parents, and all my family members for their moral support throughout my academic career. They have always helped me spur my spirits and encouraged me throughout my life with their inspiriting and hortatory words. Their prayers, patience, guidance and support lead to the successful accomplishment of this thesis.

Thanks to everybody who contributed to this achievement in a direct or an indirect way.

Contents

List of Tables

List of Figures

THESIS ABSTRACT

Name: Mohammed Moinuddin Rizwan Farooqi

Title: Adapting Tabu Search to Accommodate Online demand variations in a Data Network

Major Field: COMPUTER ENGINEERING

Date of Degree: May 2010

With the growth of the Internet, the Internet traffic increases, burdening the available network resources. The Internet service providers (ISPs) that maintain the networks try to lessen the load on the resources by implementing new technologies, by efficient utilization of the available resources, or both. The ISPs ensure proper routing of the packets resulting in effective utilization and a reduced load on their resources. Open Shortest Path First (OSPF) is the most popular and most widely used protocol for routing in the intra domain networks. OSPF selects the path of the packet based on the weight of the links. The OSPF Weight Setting problem (OSPFWS) is to find a set of weights that when applied to the links, given a traffic demand, provide a better network performance. OSPFWS is an NP-hard problem. An approach to the OSPFWS using Tabu Search (TS) already exists. The TS approach focusses on generating optimized solutions based only on the traffic demand, network topology, and link capacity. However, the solutions generated take a lot of time to obtain and are applied offline. Therefore, an approach to OSPFWS is proposed which can be used online. The objective of the thesis is to investigate and implement different online simulation system approaches using Tabu Search. To study the effects of assigning solution weights of different demands to a given demand at hand and to improve these weights to suit this demand also contributes towards the thesis work. To formulate a deterministic approach that improves the existing initial solution weights to suit the demand at hand is also part of this thesis work.

Keywords: *Traffic Engineering, Routing, OSPF, Shortest path, Utilization, OSPFWS, Tabu Search, Online Simulation, Deterministic Approach*

MASTER OF SCIENCE DEGREE

King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia

May 2010

أطروحة الملخص

الاسم : محمد موينودين ريزوان فاروقي

العنوان : تكييف تابو البحث على الانترنت لاستيعاب التغيرات في الطلب على شبكة البيانات

الرئيسية الميدانية : هندسة الحاسب الآلي

تاريخ الدرجة العلمية : 2010 يونيو

مع نمو شبكة الإنترنت ، وزيادة حركة المرور على الإنترنت ، وإثقال كاهل شبكة الموارد المتاحة ، ومزودي خدمات الإنترنت) التي تحافظ علىالشبكات في محاولة لتخفيف العبء على الموارد عن طريق تنفيذ التكنولوجيات الجديدة ، من خلال الاستخدام الفعال للموارد المتاحة ، أو على حد سواء. مقدمي خدمات الإنترنت وضمان التوجيه السليم من الحزم مما يؤدي إلى الاستخدام الفعال وتحميل انخفاض على مواردها. فتح أقصر مسار أولا (سبف) هو الأكثر شعبية والأكثر استخداما على نطاق واسع لتوجيه بروتوكول في مجال الشبكات البينية. سبف تحديد مسار الحزمة على أساس الوزن للروابط. في وزن مشكلة إعداد سبف (OSPFWS) هو العثور على مجموعة من الأوزان التي عند تطبيقها على وصلات ، ونظرا لطلب حركة المرور ، وتوفير شبكة تحسين الأداء. OSPFWS هو الثابت مشكلة الحزب الوطني. مدخلا لاستخدام OSPFWSطريقة البحث المحظور (تي اس) موجود بالفعل. ومع ذلك ، فإن نهج اتس هو أكثر تركيزا على توليد الحلول الأمثل تستند فقط على حركة الطلب طوبولوجيا شبكة ، ،وربط القدرات. وهكذا ولدت الحلول تأخذ الكثير من الوقت للحصول على وتطبق حاليا. ولذلك ، لOSPFWS يقترح نهجا والتي يمكن استخدامهاعلى الانترنت. والهدف من رسالتي هي للتحقيق في مختلف تميز الكلية على شبكة الإنترنت ، محاكاةتيم النهج وتنفيذ وتجهيز محاكاة على الانترنت باستخدام تابوالبحث. لدراسة الآثار المترتبة على تعيين الأوزان حل لمطالب مختلفة إلىنظرا للطلب في متناول اليد ، وتحسين الأوزان التي تتناسب مع هذا الطلب يساهم أيضا في سبيل العمل أطروحة. لوضع نهج القطعية أن يحسن القائمة الأولية الأوزان حل لتتناسب مع الطلب في متناول اليد هو أيضا جزء منهذه أطروحة العمل.

كلمات البحث : *هندسة المرور ، التوجيه ، سبف ، أقصر الطرق والانتفاع بها وOSPFWS ، تابو البحث ، أخبار المحاكاة ، والقطعية النهج*

درجة ماجستير في العلوم

جامعة الملك فهد للبترول والمعادن ، الظهران ، المملكة العربية السعودية

يونيو 2010

Chapter 1

Introduction and Background

1.1 Traffic Engineering and Routing

The Internet has been growing at an unprecedented rate in terms of popularity and usage. The current estimated number of Internet users is 1,565 million, which constitute about 23.3% of the world population [1]. With the ever increasing Internet usage the Internet traffic increases multi fold accordingly. The increasing traffic of the Internet strains the available network resources. To cope up with this increase, the ISPs have to either expand the network capacity or opt for a newer technology. The newer technology comes with greater capacity at an increased price. The strain on the network resources which are of limited capacity, results in congestion, at times. The ISPs may not always afford to expand the network capacity or replace the existing resources with newer technologies. The above stated problem of

network maintenance is not simple and straight forward that either expanding the network or replacing it with newer technology will solve the problem. Internet Traffic Engineering (TE) has to maintain the network resources without compromising on the network's performance. TE consists of performance evaluation and performance optimization of TCP/IP based networks. In simple words, TE is to measure, characterize, model, and control the traffic in the TCP/IP based networks [2].

ISPs like AT&T Worldnet and AOL(America Online) provide Internet connection to millions of users. The companies usually lease bandwidth from the ISPs dedicated lines. Even though the ISPs compete among themselves for the customers, they do have a business relationship (known as peering) where they provide each other connectivity to exchange their traffic [3]. The ISPs provide Internet access from the major back-bones to the regional providers, which in turn give access to thousands of local providers. The individual users can get access throught these local providers. The IP packets traverse from the source to the destination passing from one network to another, depending on the decision made by the routers involved. Each router is connected to different routers thereby resulting in multiple paths between any source and destination [4]. The task of a router is to forward the packet from the source to the destination host along the best path. The routing process becomes more complex with the increase in the number of alternate paths that a packet can take to reach its destination host [5]. To handle the complexity at an easier level, smaller domains are created within the bigger networks allowing

decisions and operations at smaller individual domain levels. The Internet is divided into smaller domains known as Autonomous Systems (ASes). The protocols that handle the routing of the traffic between two ASes are called Exterior Gateway Protocols (EGP). The protocols used in decision making of traffic routing within an AS are called Interior Gateway Protocols (IGP).

The Border Gateway Protocol (BGP) is an example of EGP while Open Shortest Path First (OSPF) [6], Routing Information Protocol (RIP) and Interior Gateway Routing Protocol (IGRP), are used in today's Internet as IGPs [7, 8]. A router maintains a routing table that contains the possible routes a packet can take to reach the destination from the source [5]. The routing protocols use different approaches in maintaining the routing table. The routing table is updated and maintained according to the routing protocol decision. When a router receives an IP packet, it forwards it to the corresponding network depending on the entry maintained in the routing table. The routing table is indexed by IP addresses, and are compared to the IP address which is present in the destination address field of the packet header.

1.1.1 Traffic Engineering

Traffic Engineering deals with the problem of achieving efficient network resource allocation while satisfying the user requirement levels [9]. Traffic Engineering controls the routing of the traffic within the network to optimize resource utilization and

network performance. In a nutshell, it is the process of measuring, characterizing, analyzing, and controlling the network traffic with the aim of improving the network performance through the application of different techniques and approaches. IGPs generate shortest paths to forward traffic that maximizes the network resource utilization. The problem with the use of shortest paths is that there exists some links that are common in different shortest paths. Different shortest paths overlap at these common links, causing congestion at these links. The traffic routed through one of these links may exceed the link capacity resulting in congestion. While this path with such a common link is being overloaded, there exists some other longer paths in which nodes are under-utilized. These are the typical problems that are to be handled by Traffic Engineering in the Internet [9]. With the ever increasing traffic, the existing network resources are strained to their capacities and thus Internet Traffic Engineering is needed to provide smooth and efficient flow of network traffic while network resources are efficiently utilized. The two considerable areas that are related to Traffic Engineering could be classified as either traffic related or resource related. The former deals with maintaining the QoS requirements that are stated in the SLA, namely packet loss rate, time delay, and throughput. The second part requires resource consideration when achieving the goals for the first part, i.e., resource utilization. Resource utilization means ensuring that all resources are utilized properly while under-utilization or over-utilization are kept to a possible minimum. The primary goal of Traffic Engineering is to reduce congestion in the network with

a minimum over-utilization of the resources [10]. The congestion problem arises because of one of the following reasons:

1. When the resource capacity does not match the offered traffic load, i.e., over-burdening of the network resources, or

2. When the traffic routing does not utilize the resource capacity efficiently resulting in some resources being over-utilized whereas others are still being under-utilized.

The first congestion problem is straightforward and can be solved by capacity increase, congestion control algorithms, or an application of both [5]. The congestion control algorithm controls the traffic demand by concentrating on regulating the traffic using the available resources so that available resources do not get overloaded. The second congestion problem is the main realm of Traffic Engineering where improper utilization of resources is remedied. The over-utilization of some of the network resources leads to packet loss, which in turn leads to more retransmissions. The packet loss and retransmissions contribute to the congestion of the network. Reducing the over-utilization of these resources leads to successful transmission without the packet loss and retransmissions. With no packet loss, retransimission due to packet loss is not done and hence a reduction in the overhead of the network. The control of the second type of congestion is reflected in the improvement of the network performance metrics because packet loss and retransmission overheads are

minimized thereby reducing congestion and increasing throughput. The metrics are generally based on costs spent on resources and their usage vs profits gained by providing the services to the customer [10].

1.1.2 OSPF Routing Protocol

The OSPF protocol is an IGP that distributes routing information to each router within an autonomous system. OSPF was developed by the OSPF working group of the Internet Engineering Task Force (IETF). It is the most widely used and is a dynamic routing protocol. It is designed for TCP/IP based networks and utilizes multicast of packets to maintain the routing updates. OSPF has the ability to respond to the changes in the topology with fewer number of routing updates [6]. IP packets are routed based solely on the destination IP address found in the IP packet header. OSPF is based on a link-state routing protocol where each router maintains a routing table that has a complete view of the network topology [11]. Each OSPF router has an identical link-state database and it sends the information related to its local interfaces and its nearest neighbors to all the other routers by flooding [12]. A router utilizes the link-state database to construct the shortest paths from itself to all the other routers in the network based on a metric called weight [13]. A weight is a positive integer used to represent the cost of each link in the network. If multiple paths to the same destination with the same weights exist, then load balancing is used among them. The traffic meant for outside the AS is forwarded to

the boundary routers along the advertised path by the boundary routers [12]. The shortest path calculation is done using the Dijkstra's algorithm [14, 15].

The following are some of the key features of OSPF [14, 16, 17]:

1. Security: Authentication mechanisms are provided in OSPF to prevent malicious intrusion from the outside, thereby providing security to the network resources.

2. Equal cost multi path: Multiple same-cost paths are allowed and used to do load balancing in OSPF.

3. Virtual links: It allows virtual links configuration, removing topological restrictions in an AS. Virtual links are used to restore/increase connectivity of the backbone. Virtual links may be configured between any pair of area border routers having interfaces to a common (non-backbone) area. The virtual link appears as an unnumbered point-to-point link in the graph for the backbone.

4. Flexible routing metric: The routing metric is a network criteria such as delay, bandwidth, or cost; and is at the discretion of the system administrator to setup.

5. Variable-length subnet support: OSPF provides support for variable-length subnet masks by providing a network mask with the destination address in

the Link State Advertisements (LSA).

6. Multi-cast support: For systems that do not understand OSPF, multi-cast is used instead of broadcast. link State Update packets are multicast on those physical networks that support multicast.

7. Stub area support: A stub area is an area which does not receive route advertisements external to the autonomous system (AS). This reduces the size of the routing databases for the internal routers of that area. To support routers having insufficient memory, areas can be configured as stubs. External LSAs (often making up the bulk of the Autonomous System) are not flooded into/throughout stub areas. Routing to external destinations in stub areas is forwarded to a default router.

1.1.3 Dijkstra's Algorithm

Dijkstra's Algorithm also known as Shortest Path First (SPF) algorithm is the basis for OSPF operations [14]. It is used to find the shortest paths from a source node to all other nodes in a network. The working of the SPF algorithm is based on the distance $d(i)$ to each node i indicating the upper bound on the shortest path length to node i. The algorithm designates two groups of nodes, namely *permanently labeled* and *temporarily labeled*. *Permanently labeled* means that the distance is the shortest of all the other alternatives and do not require change. Whereas the *temporarily*

labelled nodes are the ones which require an update to improve the existing shortest path. The algorithm starts from a given source node s and permanently label nodes according to the distances from the node s. The node s is assigned a permanent label of zero, indicating the distance from s to s is zero. All the other nodes in the network are given a temporary label equal to ∞. At each iteration, the label of a node i is the shortest distance from the source node along a path whose internal nodes are all *permanently labeled*. Dijkstra's algorithm maintains a directed out-tree T rooted at the source node that spans the nodes with finite distance labels. The algorithm maintains this tree using predecessor indices [i.e., if $(i, j) \in T$, then $pred(j) = i$]. The algorithm maintains the invariant property that every tree arc (i, j) satisfies the condition $d(j) = d(i) + c_{ij}$ with respect to the current distance labels. At termination, when distance labels represent shortest path distances, T is the shortest path tree.

1.2 Iterative Non-deterministic Heuristics

Combinatorial optimization is usually used in problems where one has to consider several different parameters to come up with a solution. Combinatorial optimization finds the ordering or arrangement of elements that results in an optimal solution to the given problem. It is used to find a solution when one or more possible optimal solutions exist. Combinatorial optimization algorithms can be broadly classified

into deterministic and non-deterministic algorithms. A deterministic algorithm progresses toward the solution by making deterministic decisions. Therefore deterministic algorithms produce the same solution for a given problem instance everytime. In non-deterministic algorithms, the solution is found using a random approach and hence leads to a different solution every time they are applied, even for the same problem instance. An iterative heuristic starts with an initial solution to the problem and then attempts to improve the resulting cost function value. If the new solution has resulted in a cost improvement, then that solution is accepted, otherwise it is rejected and search is carried on. The heuristic that is being used is a non-deterministic iterative heuristic. OSPF weight setting problem belongs to the class of combinatorial problems that are NP-hard. Deterministic algorithms are faster than iterative procedures [18]. However, deterministic algorithms are greedy in nature, looking for the gain at each step and hence have a local view of the solution space. This results in solutions which are locally optimum and have a scope for further improvement. Iterative heuristics are not plagued with this particular problem of local optima. They differ from deterministic algorithms in their probabilistic ability to get out of the local optima. In this thesis, a Tabu Search algorithm, an iterative non-deterministic heuristic, with time constraint is used for an online system.

1.2.1 Tabu Search

Tabu Search (TS) is an elegant search technique which has its influence from AI concepts. TS is a local neighborhood search [19]. TS starts with a given solution and then a large number of neighbors are generated by random moves and among them the best solution is chosen. From the generated neighborhood solutions, the best solution is chosen with the help of an evaluator. The Evaluator is a function that consists of the objectives that have to be optimized. TS uses the previous information that is gained from the past moves in moving towards the best solution. The online Tabu search algorithm is the same algorithm but the time it takes to search the neighborhood for the best solution is reduced considerably. The TS structure is presented in the following section. The detailed discussion of the working and implementation issues of TS can be found in [19].

1.3 Online System

The conventional TE approach for the OSPF based networks is to set the link cost statically for reducing traffic on congested nodes. Due to the best effort service nature of the Internet, any change in the traffic demand is not reflected in the routing of the Internet traffic. If there is a change in the traffic demand, the weight settings of the links are not changed and as a result the packets traverse the same path that was taken earlier when there was a different traffic demand. In other

words, the traffic demand changes but the paths taken by the packets to reach the destination from the source are not changed, even when there is an increase in traffic either on part of the path or the whole path. Dynamic Traffic Engineering finds new shortest paths according to the change in traffic demands where the selected path will reduce congestion and effectively utilize the network resources [20]. Dynamic Traffic Engineering advocates the use of systems that cater to the changing traffic demand and are called online systems. The main difference between the online system and the offline system is that the online system is driven by the time as a major constraint, limiting the time period for finding the solution for the problem. The limited time constraint for the online system do provide a relaxation for the quality of the solution that the online system has to find. The online system has to look for a good solution in limited amount of time whereas the offline system has an ample amount of time to look for a better solution. OSPFWS is considered in this thesis. The aim of the thesis is to find the optimized weight setting of OSPF such that congestion can be avoided for a particular traffic demand, and the optimized weight setting of OSPF should be done under the additional time constraint for its online application. The offline or regular system uses single weight settings for the network links considering long term utilization. On the other hand, the online system considers the short term traffic fluctuations and has to find a good weight setting for this changed traffic demand before the demand changes again. So the online system has to find the new traffic demand weight solution with the

additional constraint of time. Fortz et al. [21] show that the optimization of the link weights results in routing that effectively utilizes the network resources. Fortz et al. came up with a cost function based on the range of utilization in each link. Sqalli et al. [22] have shown through experimentation that the Fortz cost function does not address the optimization of the number of congested links,and proposed a new cost function. The Companies have started applying the routing optimizations discussed above in one form or another with a proven significant benefit of substantial savings and improved efficiency in their networks [23]. The optimization approaches discussed above are developed with the aim of generating optimized solutions that have to be obtained off-line [24]. The problem of assigning weights has been adressed earlier by Sqalli et al. [22, 24] using different non deterministic heuristics. An approach to the OSPFWS using Tabu Search (TS) already exists. TS has been able to improve the results that have been generated earlier by Genetic Algorithm, Simulated Annealing, and Simulated Evolution approaches [25]. The TS approach [25] has been designed with a focus on generating optimized solutions based on the traffic demand, network topology, and link capacity. The TS approach is more focussed for working as an offline system, which takes a lot of time and generates good solution weights that will be used for link weights, disregarding the short term traffic demand fluctuations. With the changing number of Internet users at a given time, the traffic demand changes. When the traffic demand changes, the inputs to the optimization algorithms also change. With the change in the inputs, i.e.,

traffic demand, the problem changes and thereby creating a need for a different solution for the new traffic demand. This was not considered in the works done by [21, 22, 26, 27] . In this thesis, TS is applied to find the solution weights that considers the change in the traffic demand for the online system. This thesis also contributes towards a deterministic approach of changing the weights for the links so that the cost decreases.

1.4 Thesis Objectives

In this work, an NP-hard problem related to online OSPFWS is addressed and solved by the application of a non-deterministic iterative heuristic and a deterministic approach. The objective is to set the new OSPF weights on the network links such that the network is utilized efficiently even after the traffic demand changes. Therefore, an approach to OSPFWS is proposed which can be used online. The main objective of the thesis is to find the weight settings configuration that provides less congestion on the links in a limited time. Thus the main focus of this thesis work is the study of the existing methods for an online system for the OSPFWS problem. The following is the list of objectives for this thesis:

1. Investigate different strategies of proposed solutions using iterative heuristic techniques for OSPFWS online system. Investigate the existing online simulation systems and their approaches of finding the solution.

2. Investigate and study the effect of applying different stored traffic demand's weight solutions to a network with a new online traffic demand and computing the cost and maximum utilization of the network after the above assignment is made.

3. Provide a method for choosing an initial solution, i.e, weight setting for the demand at hand.

4. Design and implement an online simulation system (OLS) processing module that assigns and improves the initial weight solution that is used for the online traffic demand by adapting TS to the online system.

5. Formulate a new deterministic approach that improves the initial weight solution that is used for the online traffic demand.

1.5 Organization of Thesis

The rest of the thesis is organized as follows. Chapter 2 presents a survey of literature related to the OSPFWS problem. In particular, problems of dynamic TE and the solutions to the problem are reviewed, including the alternatives to the online processing. In Chapter 3, the online OSPF weight setting problem is formulated. The cost function that is used with the online system is also described. Chapter 4 discusses the implementation details of the proposed online system that adapts TS

to find solutions in a lesser time. The different modules of the online system are discussed as well. Chapter 5 provides details on the experimental results of adapting TS for the online system. The test cases are also described in Chapter 5. The deterministic approach for the online system is discussed in details in Chapter 6. The experimental results of the deterministic approach of online system are detailed in Chapter 7 as well. Chapter 8 provides conclusions and possible future directions of work.

Chapter 2

Literature Review

2.1 Introduction

Combinatorial optimization is usually used in problems where one has to consider several different parameters to come up with a solution. OSPF is the most used protocol for routing in the intra domain networks [6]. OSPF selects the path of the packets based on the weight of the links. The OSPFWS problem is to find a set of weights that when applied to the links, and for a given traffic demand, provide a better network performance. The OSPFWS problem belongs to the class of combinatorial problems that are NP-hard and hence non-deterministic iterative heuristics are applied. GA, Sa [24], SimE [22] and TS [25] have been applied to find the optimized weight settings for a given demand which results in a reduced cost and improved performance of the network. In the following paragraphs a review

of various relevant topics, concepts, and solution approaches related to OSPFWS is provided. Review of the solution approaches to dynamic TE including online systems approaches is given. The Tabu Search algorithm with time constraint is also discussed.

2.2 Tabu Search (TS)

TS is a local neighborhood search which is used in solving combinatorial optimization problems of diverse fields [19]. The search starts with an initial feasible solution and searches in the neighborhood of the solution by doing a sequence of random moves or perturbations. The recent moves and their characteristics are stored in a tabu list until the solution is found or a stopping criteria is met. The purpose of this list is to ensure that the search is not done in the areas that have been explored before and thus preventing it from remaining or entering in the local optima. In every iteration, a probable solution from the neighborhood is generated and is maintained in the candidate list, and this probable neighborhood solution can be achieved by making a certain number of moves from the current solution. The best move that would result in the best solution is accepted only if it is not in the tabu list. If the best move found is in the tabu list, the best solution is checked against an aspiration criterion and if satisfied the move is accepted. The aspiration criterion takes precedence over the restrictions in the tabu list. In certain conditions, a move that

is already in the tabu list is accepted for the simple reason that it may take the search towards a new region because of the intermediate moves.

The tabu list size and the chosen aspiration criterion are responsible for the TS behavior. The search can be intensified or diversified because of short-term, intermediate-term, and long-term memory components that are of direct relation with the tabu list size. The major idea of the short-term memory component is to classify certain search directions as tabu (or forbidden) so as to avoid returning to previously visited solutions. The TS that is being discussed in this thesis is with short-term memory component. The goal of the long-term memory component is to diversify the search into new regions that are different from those examined thus far. The role of the intermediate-term memory component is to intensify the search and make it more aggressive. A selected number of best trial solutions generated are chosen and their features are recorded and compared. The search is done for the solutions that contain these features.

The goal of the tabu list is to not allow the moves that are in it and hence restrict the tabu search. In other words, a tabu list maintains the restrictions of the search and the aspiration criterion helps the search in overriding these restrictions. This means that if a tabu move is found which is in the tabu list but also happens to meet a certain aspiration criterion then that move is accepted even if the tabu list restricts this particular move. This move is accepted in order not to get stuck with the local optima and look for better solution beyond these neighborhood. The

online Tabu Search algorithm is the same as the TS algorithm, but the time it takes to search the neighborhood for the best solution is reduced considerably. The main modification of the TS to adapt it for the online system is done by providing TS with an initial solution instead of a random one. The cost of the random solution used by the TS is far from being optimum. The initial solution that is used by the Online TS is the optimum solution generated by the Offline TS for a different demand. The results generated by the Offline TS for the different demands are stored. These stored results are used as the starting point by the Online TS to look for a better result for the changed demand. The weights for traffic demands D4, D6, D8, D10, D11, and D12 are generated using Offline TS. These weights generated by the Offline TS result in a reduced cost for the respective demands. The Online TS uses these weights as the initial solution for the new changed traffic demand DR. All of the optimal solution weights of the traffic demands D4, D6, D8, D10, D11, and D12 are used as the initial solution from which the TS looks for the optimal solution for DR. The TS is applied once the initial weight solutions for the demand are assigned and the cost computed. Then, TS looks for better weight solutions that would further reduce the cost of the network if these weight solutions are applied on the links of this network with demand DR. The TS structure is presented in the following section. The detailed discussion of the working and implementation issues of TS can be found in [19].

2.3 Tabu Search (TS) Algorithm

Ω : Set of feasible solutions.
S : Current solution.
S^* : Best admissible solution.
$Cost$: Objective function.
$\aleph(S)$: Neighborhood of $S \in \Omega$.
$\mathbf{V}^*$: Sample of neighborhood solutions.
T : Tabu list.
AL : Aspiration Level.

Begin
Start with an initial feasible solution $S \in \Omega$.
Initialize tabu lists and aspiration level.
For fixed number of iterations **Do**
    Generate neighbor solutions $\mathbf{V}^* \subset \aleph(S)$.
    Find best $S^* \in \mathbf{V}^*$.
    If move S to S^* is not in **T** **Then**
        Accept move and update best solution.
        Update tabu list and aspiration level.
        Increment iteration number.
    Else
        If $Cost(S^*) <$ **AL Then**
            Accept move and update best solution.
            Update tabu list and aspiration level.
            Increment iteration number.
        EndIf
    EndIf
EndFor
End.

Figure 2.1: Algorithm : Tabu Search (TS).

2.4 Related work

OSPF is an IGP used for routing within the autonomous systems and uses Dijkstra's algorithm for calculating the shortest paths within an AS [17]. OSPF is a link-state routing protocol that sends link-state advertisements (LSAs) to all other routers within the same hierarchical area. As OSPF routers gather link-state information, the SPF algorithm is used to calculate the shortest path to each node from within the network [7]. The link weight is used to decide about the shortest path a packet has to take. The selection of link weights should be done such that congestion is to be avoided in the network [23]. Fortz and Thorup [26] have shown that optimization of the link weights so as to guarantee congestion avoidance and efficient network utilization is an NP-hard problem.

Fortz and Thorup [26] proposed a local search heuristic and implemented it for the AT&T backbone network along with different other synthetic networks. The results in [26] were improved by applying local search heuristic in [27]. A new cost function formulated by Sqalli et al. considered utilization and the extra load on the congested links in the network for optimization. This resulted in fewer congested links and thereby reducing the overall network utilization [22]. Dynamic Traffic Engineering involves solutions that should consider the new traffic demand and processing time constraint. These solutions should be better than the existing solution and need not be the best possible solution for the changed demand. In [28], Ivan

et al. have proposed an Adaptive Multi Path algorithm (AMP) for dynamic Traffic Engineering to reduce the congestion in the network with the changing traffic demands. The AMP is a multi path routing algorithm of the OSPF Optimized Multipath (OSPF-MP) class. These multi path routing algorithms involve informing all routers in the network about the load of every link. Then, the routers depending on this information move traffic from congested to less congested paths. The OSPF-MP class algorithms require additional data structures and contribute to the traffic with the link congestion information. The AMP is able to reduce the signaling overhead traffic and memory consumption compared to the rest of OSPF-MP algorithms [28]. In contrast to the related multipath routing algorithms, AMP does not employ a global perspective of the network in each node. The AMP restricts available information to a local scope that results in reducing signaling overhead and memory consumption [28]. AMP does not require that each node has to know about all the other paths in the network, thereby reducing memory requirement. As AMP provides nodes only with local network information about the links to its direct neighbors, the signal overhead is also reduced. AMP tries to find multiple alternate paths so that a secondary route could be used and the congested route avoided. AMP is successful in alleviating the two drawbacks, i.e., signal overhead and memory structure [28]. The traditional routing protocols are designed for achieving network robustness and are not capable of adjusting the routes in case of changing traffic demands. The current Internet routing architecture uses numerical

values assigned to links, and are called link costs. These link costs are the basis for the calculation of network paths. Paths between any two nodes in the domain are determined by minimizing the sum of link costs over all path candidates. If there are two paths with two different link costs, the path with the lower link cost would be chosen over the higher one. Simply stated, to make sure that a link is selected in the path, the cost of that link should be lesser than the other alternative links. This means that the selection of the path depends indirectly on the appropriate setting of link cost. In the majority of the ISPs networks, the link cost values are maintained unchanged for several days. During those days the traffic always takes the same path from source to destination. In the event of network congestion, the traffic will still be routed over the congested links [20]. The routing over congested links is done because the link cost had been set earlier and the paths have been calculated before the congestion. This results in a reduction of network efficiency. In order to solve this problem, many approaches have been developed. Multiprotocol Label Switching (MPLS) [29] is basically a technology for establishing virtual circuits between any pair of routers in a network domain [30, 8]. MPLS requires extensive network management requirements for allocating network resources to individual paths and path maintenance. In contrast to the above mentioned MPLS which is centrally managed, decentralized approaches also exist. In a decentralized approach, link costs are dynamically adjusted in proportion to the instantaneous link delays. This scheme performs well under low and medium loads, but leads to link load oscillations

for high load. The traffic changes are uneven through out the network and because of the traffic changes the link weights are changed as well. The changed cost of the links at which traffic has been changed affects other paths that involve links with no traffic change by increasing the overall cost for this path. A single cost change affects multiple paths at the same time. Thus, under heavy loads and with many single cost changes, we get link load oscillations [13]. The OSPF-MP protocol aims at achieving optimal load distribution automatically and dynamically, using a link state protocol flooding mechanism for informing all routers in the network about the load of every link. Equipped with this information, the routers can shift traffic from congested to less congested paths and reduce maximum link utilization in the network [31]. However OSPF-MP protocols require each node to have knowledge of all the other network paths and overhead in transferring this data on the network from each node to every other node.

Dynamic Traffic Engineering also employs link congestion avoidance. Link congestion avoidance is employed because link weights used are static and are set considering long term congestion reduction. This leads to the neglect of short term traffic changes and hence results in temporary performance degradation. Over a long period of time, many occurences of similar temporary performance degradation happen but are not considered. The OLS system consider these temporary performance degradations in their problem solving approach. Online systems such as Online Simulation (OLS) system [32] and Online Simulation (OLS) framework

[33] are used to monitor and generate optimized solutions according to the traffic changes. The OLS system of [32] measures the network performance, predicts the network condition, and reconfigures the network parameters according to the simulation results. The OLS system used only 2 simulation results in concluding the new parameters for the network. The testbed used in [32] has only 4 routers and 8 nodes and the number of simulation scenarios used is limited to two only. The online system that is being proposed here will be simulating a more complex and large scale real network. As the number of nodes are increased in a network, the change in one of the links weight would affect the other paths link weights as well, and hence complicate the problem of finding weight solution for the given demand at hand. The above scenario would resemble the real problems faced in large scale networks and hence would mean that the proposed online system would be more suitable in finding the solution for the real networks than the approach outlined by Tamura et al. [32]. Kaur et al [33] uses a Recursive Random Search (RRS) technique and has performed simulations for a single large network with 48 nodes only. Hema et al. in [33] have used the number of function evaluations to show how fast RRS was able to find a "good" weight setting when compared to other techniques, and has not shown how much they were able to improve the network performance after the change in demand traffic. The term "good" weight setting is not discussed furthermore and no comparison on the solution is detailed. The number of function evaluations is used as a metric to prove that RRS finds the solution faster than the

local search heuristics used by Fortz et al. [27].

This thesis contributes towards the OLS processing model that involves an iterative heuristic, namely TS, to find an optimal solution for the demand weights in a limited time frame. The objective of this thesis is to implement the online simulation processing using Tabu Search. The study of the feasibility of assigning the weight solution of different standard demands to a given demand at hand D (mimicking the changed traffic demand), as an initial solution is also part of the thesis work.

Chapter 3

Problem Description and Cost Functions

3.1 Problem Statement

The following subsection provides details of notations used to formulate the OSPF weight setting problem considering cost and maximum utilization.

3.1.1 Notation

G Graph.

N Set of nodes.

n A single element in set N.

A Set of arcs.

A^t	Set of arcs representing shortest paths from all sources to destination node t.
a	A single element in set A. It can also be represented as (i, j).
s	Source node.
v	Intermediate node.
t	Destination node.
D	Demand matrix.
$D[s, t]$	An element in the demand matrix that specifies the amount of demand from source node s to destination node t. It can also be specified as d_{st}.
w_{ij}	Weight on arc (i, j). If $a = (i, j)$, then it can also be represented as w_a.
c_{ij}	Capacity on arc (i, j). If $a = (i, j)$, then it can also be represented as c_a.
Φ	Cost function.
$\Phi_{i,j}$	Cost associated with arc (i, j). If $a = (i, j)$, then it can also be represented as Φ_a.
δ_u^t	Outdegree of node u when destination node is t.
$\delta^+(u)$	Outdegree of node u.
$\delta^-(u)$	Indegree of node u.

l_a^t Load on arc a when destination node is t.

l_a Total load on arc a.

$f_a^{(s,t)}$ Traffic flow from node s to t over arc a.

3.1.2 Assumptions and Terminology

1. A single element in the set N is called a "Node".

2. A single element in the set A is called a "Arc".

3. A set $G = (N, A)$ is a graph defined as a finite nonempty set N of nodes and a collection A of pairs of distinct nodes from N.

4. A "directed graph" or "digraph" $G = (N, A)$ is a finite nonempty set N of nodes and a collection A of ordered pairs of distinct nodes from N; each ordered pair of nodes in A is called a "directed arc".

5. A digraph is "strongly connected" if for each pair of nodes i and j there is a directed path $(i = n_1, n_2, ..., n_l = j)$ from i to j. A given graph G must be strongly connected for this problem.

6. A "demand matrix" is a matrix that specifies the traffic flow between s and t, for each pair $(s, t) \in NXN$.

7. $(n_1, n_2, ..., n_l)$ is a "directed walk" in a digraph G if (n_i, n_{i+1}) is a directed arc in G for $1 \leq i \leq l - 1$.

8. A "directed path" is a directed walk with no repeated nodes.

9. Given any directed path $p = (i, j, k, ..., l, m)$, the "length" of p is defined as $w_{ij} + w_{jk} + ... + w_{lm}$.

10. The "outdegree" of a node u is a set of arcs leaving node u i.e., $\{(u, v) : (u, v) \in A\}$.

11. The "indegree" of a node u is a set of arcs entering node u i.e.,$\{(v, u) : (v, u) \in A\}$.

12. The input to the problem will be a graph G, a demand matrix D, and capacities of each arc.

3.1.3 Problem Formulation

The OSPF weight setting (OSPFWS) problem can be stated as follows: Given a network topology and predicted traffic demands, find a set of OSPF weights that optimizes network performance. More precisely, given a directed network $G = (N, A)$, a demand matrix D, and capacity C_a for each arc $a \in A$, we want to determine a positive integer weight $w_a \in [1, w_{max}]$ for each arc $a \in A$ such that the objective function or cost function Φ is minimized. w_{max} is a user-defined upper limit. The chosen arc weights determine the shortest paths, which in turn completely determine the routing of traffic flow, the loads on the arcs, and the value of the cost

function Φ. The quality of OSPF routing depends highly on the choice of weights. Figure 1 shows a topology with edge numbers and assigned weights within the range $[1, 20]$. In this case, in ascending edge number sequence, i.e., 1 2 3 ... 14, the solution can be specified as:

Solution=$(1, 5, 18, 13, 4, 15, 18, 17, 7, 16, 19, 16, 14, 3)$.

The above solution lists the weights that are for the following edge numbers: **Edge numbers**=$(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14)$.

3.2 Mathematical model and Cost Function

Using the notations described in the earlier section, the problem can be formulated as the following multi-commodity flow problem [22, 34]. For detailed cost analysis, refer to the work done by Fortz et al in [22, 34].

$$minimize \quad \Phi = \sum_{a \in A} \Phi_a(l_a)$$

subject to these constraints:

$$\sum_{a \in \delta^+(u)} f_a^{(s,t)} - \sum_{a \in \delta^-(u)} f_a^{(s,t)} = \begin{cases} -D(s,t) & \text{if } v = s, \\ D(s,t) & \text{if } v = t, \\ 0 & \text{otherwise,} \end{cases} \qquad v, s, t \in N, \qquad (1)$$

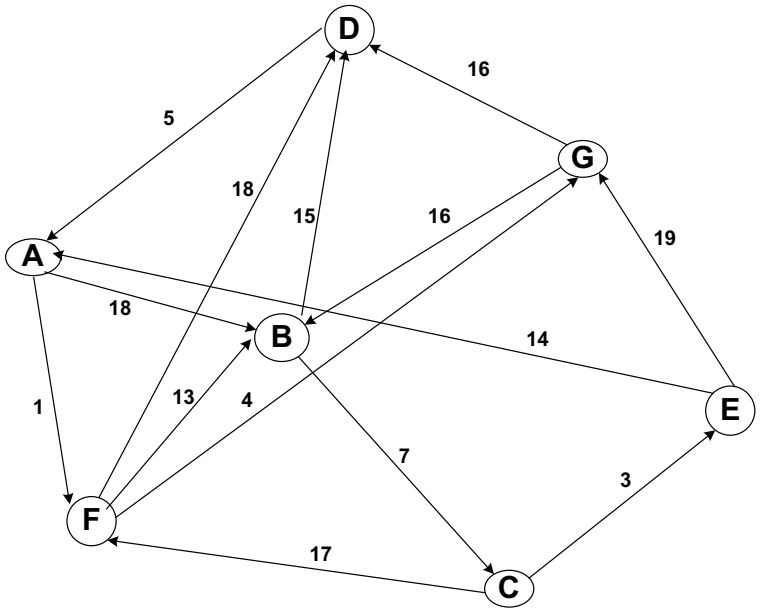

Figure 3.1: Representation of a solution to an OSPF weight setting problem.

$$l_a = \sum_{(s,t)\in NXN} f_a^{(s,t)} \qquad a \in A, \qquad (2)$$

$$f_a^{(s,t)} \geq 0 \qquad (3)$$

The constraints listed above are taken directly from the work done by Fortz et al in [22, 34].

Constraints (1) are flow conservation constraints that ensure that the desired traffic flow is routed from s to t, and constraints (2) define the load on each arc. As Φ is a convex objective function and all constraints are linear, this problem can be solved optimally in polynomial time [27].

In experiments, Φ_a are piecewise linear functions, with $\Phi_a(0) = 0$ and derivative, $\Phi_a^{'}(l_a)$ given by:

$$\Phi_a^{'}(l) = \begin{cases} 1 & \text{for } 0 \leq l/c_a < 1/3, \\ 3 & \text{for } 1/3 \leq l/c_a < 2/3, \\ 10 & \text{for } 2/3 \leq l/c_a < 9/10, \\ 70 & \text{for } 9/10 \leq l/c_a < 1, \\ 500 & \text{for } 1 \leq l/c_a < 11/10, \\ 5000 & \text{for } 11/10 \leq l/c_a < infinity \end{cases} \tag{4}$$

3.3 Formulation of the Cost Function

In this section, detailed explaination of the steps to compute the cost function Φ for a given weight setting $\{w_a\}_{a \in A}$ and a given graph $G = (N, A)$ with capacities $\{c_a\}_{a \in A}$ and demands $d_{st} \in D$ are given. This procedure is also described in [26].

A given weight setting will completely determine the shortest paths, which in turn determine the OSPF routing, and how much of the demand is sent over which arcs. The load on each arc gives us the link utilization on this arc, which in turn gives us a cost using the function Φ_a. The total cost Φ is the sum of the individual costs Φ_a where $a \in A$.

The basic problem is to compute the arc loads l_a resulting from the given weight setting $\{w_a\}_{a \in A}$. The arc loads are computed in five steps. For all demand pairs $d_{st} \in D$, consider one destination t at a time and compute partial arc loads $l_a^t \; \forall \; t \in \bar{N} \subseteq N$, where $\bar{N}$ is the set of destination nodes.

1. Compute the shortest distances d_u^t to t from each node $u \in N$, using Dijkstra's shortest path algorithm [35]. Dijkstra's alogrithm usually computes the distances away from source s, but since we want to compute the distance to the sink node t, the algorithm will be applied on the graph obtained by reversing all arcs in G.

2. Compute the set A^t of arcs on shortest paths to t as,

$$A^t = \{(u, v) \in A : d_u^t - d_v^t = w_{(u,v)}\}.$$

3. For each node u, let δ_u^t denote its out degree in $G^t = (N, A^t)$, i.e.,

$$\delta_u^t = \mid \{v \in N : (u, v) \in A_t\} \mid$$

   If $\delta_u^t > 1$, then traffic flow is split at node u to balance the load.

4. The partial loads l_a^t are computed as:

   (a) Nodes $v \in N$ are visited in order of decreasing distance d_v^t to t.

(b) When visiting a node v, for all $(v, w) \in A^t$, set

$$l^t_{(v,w)} = 1/\delta^t_v(d_{vt} + \sum_{(u,v)\in A^t} l^t_{(u,v)})$$

5. The arc load l_a is now summed from the partial loads as:

$$l_a = \sum_{t\in\bar{N}} l^t_a$$

The evaluated costs are normalized to allow us to compare costs across different sizes and topologies of networks.

3.4 Normalization of the Cost Function

The normalization of cost is given in [26]. Cost is normalized to keep it independent of the network topology and demand matrix. Let Ψ be the cost if all the flow was sent along the hop-count shortest paths and the capacities matched the loads. Let $\Delta(s, t)$ be the hop-count distance between s and t. Ψ is calculated as,

$$\Psi = \sum_{(s,t)\in NXN} \ (32/3 \ . \ D[s, t] \ . \ \Delta(s, t))$$

The normalized cost function is,

$$\Phi^* = \Phi/\Psi$$

3.5 Online System Problem Formulation

In this section, the online system problem is described. Given a network topology and a new traffic demand, find a set of OSPF weights that optimizes network performance. The OSPF weights should be found within a certain small time interval. The weight solution of another traffic demand is provided as an initial point. In other words, given a directed network $G = (N, A)$, capacity C_a, and initial weight Wi_a for each arc $a \in A$, we want to determine a positive integer weight $w_a \in [1, w_{max}]$ for each arc $a \in A$ such that the objective function or cost function Φ is minimized for the new traffic demand matrix D, and this has to be done within a small time interval T.

The online system comprises of three modules, namely Monitor, Improve, and Configure. The Monitor module simulates the change in the traffic demand and triggers the Improve module with the changed traffic demand matrix D. The problem of finding weight solutions described in the start of the section is handled by the Improve module. The Improve module has to find the initial weight solution Iw_a that belongs to the same directed network $G = (N, A)$, with the same capacity C_a for each arc $a \in A$ but with a different traffic demand matrix ID. When the Improve module has found the initial weight solution Iw_a with the different traffic demand matrix ID, an attempt to improve these weights Iw_a for the given traffic demand matrix D within the time interval T is done. Once the Improve module has found

the weight settings for the new traffic demand matrix D within the time interval T, the results are passed to the Configure module. The Configure module is assumed to apply the weights on the links in the network. The Improve module is implemented with the Online TS and the Deterministic Approach.

The problem of finding weights for the changed traffic demand in the network using one of the standard demand weights as an initial solution within a limited time is solved in this thesis.

Chapter 4

Online System

Section 4.1 describes the dynamic TE and the Online Simulation (OLS) concepts. Section 4.2 discusses the terms, definitions, and their relation with each other. The three modules of the online system and their implementation details are presented in section 4.3. The limitations and challenges of the OLS are discussed in section 4.4. The OLS algorithm is presented in section 4.5. The Online TS algorithm is described in section 4.6. The differences between the Online and the Offline TS algorithm are also listed in this section.

4.1 Introduction and Motivations

Dynamic Traffic Engineering advocates the use of systems that cater to the changing traffic demand and are called online systems. The goal of dynamic Traffic Engineer-

ing is to respond to the network parameters such as traffic demand to ensure better operation of the network that fits the current network situation. In dynamic Traffic Engineering, load balancing is achieved by changing the link weight depending on the local traffic conditions. The load balancing achieved by changing the link weight depending on the local traffic conditions is called adaptive routing [28].

The problems associated with adaptive routing or traffic sensitive routing are [20]:

1. The frequent route changes that need to be done when every local traffic condition is considered results in an unstable system.

2. It considers local traffic changes and does not consider the state of the complete network, thereby it cannot achieve optimization in the complete network resource utilization.

To overcome the above problems, an Online Simulation (OLS) system is used for dynamic Traffic Engineering. The first problem is taken care of by maintaining a lower bound on the time period of the traffic change, i.e., not every change in traffic demand is catered with new weight settings. The OLS does not only work with the local traffic conditions, but also considers the whole network's traffic demand. This is a common assumption in all the TE work at the intra-domain level.

4.2 Terms, Definitions, and Relations

The main difference between the online system and the existing offline system is the time spent in finding the solution weights for the network with the changed demand. The offline system has no restriction of time whereas the online system has very limited time to reach an acceptable solution. The time spent by the online system in finding a good OSPF weights solution for the network is called the processing time. In other words, the time that is required for finding the new weight settings for the network because of the changed traffic demand is called **processing time**.

The Period of Traffic Demand Change: The online system continuously monitors the network and has to find the solution weights for the demand changes in the network. These demand changes are unpredictable. Every network demand change is not catered to, instead the demand change within a time period is looked after. If every demand change is responded to, then the system will not generate favorable results because the trigger will not allow any time gap to update the new found weights. To understand this scenario, let us discuss it through an example. Let the time starts at 't', and we will be discussing the scenario for time 't' onwards. Let us say that there is a demand change at the start of the time 't'+5 minutes and the online system has been triggered and the processing module starts looking for the solution with the new network demand change. The processing time spent in finding the new weights is, e.g., 5 minutes. The processing module generates the

new weights for the network in 5 minutes. Now the time from the start is 't'+10 minutes and the update of the weight changes has to be applied. Let us say that the time spent in updating the new weights is 2 minutes. That gives us the latest time of 't'+12 minutes. In this time period from 't' to 't'+12 minutes, the network has seen the change in the traffic demand and as a result it triggered the online system and the processing of the new weights. The update is also done in the time span of 12 minutes. For example, another change in the traffic demand at 't'+10 minutes happens, while the processing of the earlier traffic demand change is still underway. This new change in the traffic demand triggers the online system and the processing module starts finding the new weights. After 't'+15 minutes the online system comes up with the new weights. The solution weights that have been found for the first traffic demand change in the network had been updated at 't'+12 minutes would be changed at 't'+17 minutes. The new weights were used only for 5 minutes and were changed again. In this scenario, the new weights have been changed in just 5 minutes. Consider another scenario in which the network traffic demand changes at 't'+11 and at 't'+12 minutes. Then, the new weights that cater to the demand change at 't'+11 are updated at 't'+18 minutes (processing time = 5 minutes and update time = 2 minutes) and the very next minute the new weights associated with the traffic demand at 't'+12 minutes change. This example helps in understanding that every traffic demand change is not fruitful to attend to. It is better to cap the time for the traffic demand change so as to utilize the 5 minutes processing time

spent, otherwise the processing time of 5 minutes is wasted in finding new weights which will be replaced within one minute by the other new weights (corresponding to the more recent traffic demand change).

Convergence time : Once the processing module of the online system finishes finding the OSPF weights, it forwards them for the implementation module to apply these new weight settings. The new weights are sent to the nodes in the network and the nodes recompute their tables for the OSPF link costs. The time taken by the network to implement the new node weights and update the link cost tables with the newly computed paths is called the convergence time. The convergence time in OSPF networks in the experimental setup used in [35] is approximately 15 seconds.

Update time : This is the amount of time taken by the online system to apply the new weight solutions to the nodes in the network. The update time is the time required to update of the new weight solutions to the corresponding links.

The Update time and the Convergence time are two different measures. The Update time is concerned with only assigning the new weight solutions to the links. The Update time is the time needed by the new weight solutions to reach the nodes, whereas the Convergence time is the time taken by the network nodes to recompute their paths and path costs in the routing tables. In other words, the time taken by the network to converge on the new routes and their cost is called Convergence time. The Update time is very much greater than the Convergence time. The Convergence time in OSPF is in the order of few seconds, typically 15 seconds [35], while

the Update time is in minutes.

Overhead: It is the extra time spent in processing, which is more than the time spent for applying the updates. In other words,

$Overhead = processingtime/Updatetime.$

The goal of the online system is to maintain the overhead at its lowest, so that the system does not spend more time in finding the weights rather than using them.

The processing time is dependent on other factors such as Update time, Period of Traffic Demand Change, Convergence time, and the Overhead. The online system has to do the processing in a limited time. The Processing time given for the online system to find the solution and the quality of the solution it reaches are dependent on one another. If the processing time is increased and the online system is allowed to look for the solution within this increased time then the quality of the solution it finds in the increased time would be better than the quality of the solution obtained within a shorter time. The quality of the solution and the processing time of the online system are directly proportional. Figure 4.1 is plotted for the best cost in function of the time. The best cost is on the Y-axis and the time taken for the processing is on the X-axis. Figure 4.1 shows the relation between the best cost and the processing time for the Online system. If the processing time is increased, it means that the quality of solution increases thereby resulting in fewer number of updates. Fewer number of updates means an increase in the time interval between two consecutive updates.

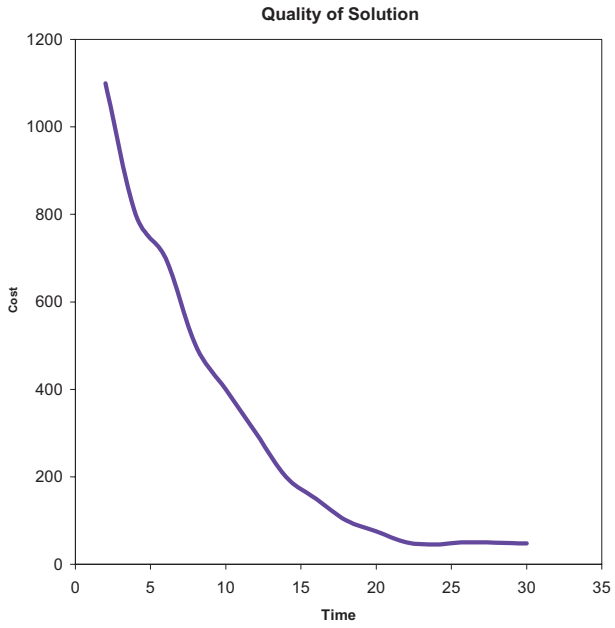

Figure 4.1: Best Cost Versus Processing time in an Online System.

4.3 Modules

The online system proposed works with the traffic demand matrix for the complete network as an input and looks for a solution that when applied improves the complete network's performance.

The concept of the OLS system is depicted in the Figure 4.2. The database of the solution weights is the collection of the solutions generated by the Offline Tabu search. The sorting process of the database is done according to the network topology and the traffic demand matrix for which the solutions were generated.

The OLS (systems) comprise of three modules, in general, as follows [32] [33]:

1. Monitor: The job of this module is to monitor the network performance based

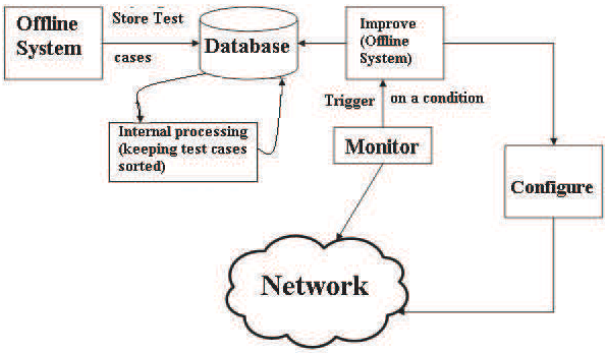

Figure 4.2: OLS Design Concept.

on different network performance metrics. The monitor model is assumed to work and is not implemented.

2. Improve: The Monitor module submits input to this module. The Improve module takes the input traffic demand from the Monitor module along with the network topology and the resource capacity, and then finds the optimized weight settings to improve the resource utilization and reduce the cost.

   The Improve module is the essence of the online system because it has to do all this processing within a real time constraint and its results ensure that optimization is achieved. The Improve module is implemented in this thesis.

3. Configure: Once the Improve module is finished with its work and has come up with the new weight settings, the new weight settings are passed to this module. The job of this module is to configure the new weight settings on the links. In this thesis, the Configure module is not implemented.

4.4 Limitations and challenges

The Online Simulation (OLS) system is used for dynamic Traffic Engineering. Online systems are required to respond to the short term traffic fluctuations and to find an optimized solution weights for the network resources. The main difference between the online and offline systems is the time constraint and the quality of the solution. The offline system of finding weight settings for OSPF has ample time to find the solution weights such that they minimize the cost and resource utilization. The online system on the other hand has to come up with the OSPF solution weights in a very short time (compared to the time offline systems take), i.e., the online system has to be a lot faster. The online system has to be fast on one hand whereas on the other it does enjoy a relaxed goal of cost and utilization reduction with comparison to the offline system. If the simulation time is increased, it would result in better solutions but the time has to be limited to satisfy the constraint of an online system. Hence, the quality of the solution is relaxed to limit the simulation time.

The online system needs to come up with a solution that is better than the existing solution. The word marginally is loose and has to be viewed as a comparative term. The comparison refer to the gain achieved in terms of network performance if the new settings are applied to the effort applied in terms of traffic overhead generated in setting the new weights on the links in the network. The effort refers to the time

spent in sending LSAs so that the OSPF converges and comes up with new shortest paths for all the routers. The online system searches for a weight solution for all the network links and is not restricted to a local set. The OLS does not limit itself to the local traffic conditions only, but rather considers the whole network's traffic demand and hence is able to optimize the performance of the network. The OLS is triggered by the over-utilization of the network resources. In this work, the triggering of the OLS is assumed. It then starts it search with a given initial solution and looks for a best solution according to the traffic demand. The online system simulates the network conditions and tries to come up with the best possible solution within the time required.

The offline system uses TS to look for a better solution for the OSPFWS. The offline system generates a solution for each network topology spending almost one hour of time to reach an optimal solution for each of the network standard traffic demands. These solutions that have been gathered are specific to the given traffic demands. The offline system obtains this solution by starting from a random initial solution and then searching the neighborhood space to get the optimal solution. However, the Online Simulation system improves on that by taking the already generated optimal solution for the standard traffic demands as the initial solution and then traversing the neighborhood space for an optimal solution for the new given traffic demand. The initial solution (optimal solutions of standard traffic demands) gives the online system a good potential solution space as a starting point to search on-

wards. To model a new traffic demand a random demand has been used which is different from all the existing standard traffic demands. The randomness of this traffic demand helps in simulating a demand change in the network. The existing standard demands are different for each network topology and hence the random traffic demand created is different from these standard traffic demands as well.

The core of the OLS is the Improve module or the processing module. The terms Improve and processing are used interchangeably throughout the thesis to indicate the process of finding better weight solutions to suit the new traffic demand. The Monitor module is responsible for triggering the processing module. But, since the processing module is simulated, this trigger is not implemented. The Configure module is assumed to assign the weights to the links and hence has not been implemented. The monitor and the configure module are not implemented. The method of choosing the initial solution (out of all the stored demand optimal solutions) is based on the observation of which stored optimal demand solution serves as best initial solution from the generated results. The demands values that are present in a demand file are added up to give a total value of the demand. For instance, D4 is smaller than D6 and D8. The demand which is to be used as the initial solution by the OLS system is the one which is the closest to the new traffic demand. The time duration for which the processing of the solution is done in the Online TS is of 360 seconds. This time frame can also be reduced, and the gain in the cost of the solution could still be observed.

4.5 Online Simulation (OLS) Algorithm

START

1. Get input network graph to simulate a network;
2. Get input network resource capacity and store capacity for cost and utilization calculation of the simulated network;
3. Get input network traffic demand and store traffic demand. This step simulates the change in the network traffic demand and simulates the triggering of the online system. In a real setup, the triggering will be done by the Monitor module of the online system;
4. Get initial solution weight settings. The Monitor module is assumed to provide the Improve module with the initial weight setting from the database;
5. Compute shortest path;
6. Calculate fortz cost with initial weight solution settings to the given traffic demand;
7. Calculate maximum utilization and number of congested links with the initial solution weight settings applied;
8. Call tabu search to find weight solutions from the neighborhood of the initial solution to improve the cost, maximum utilization, and number of congested links;
9. The results are ready to be updated by the configure module (which is not implemented here);

End

Figure 4.3: Algorithm : Online Simulation Algorithm (OLS).

4.6 Online Tabu Search (TS) Algorithm

The Online Tabu Search (TS) algorithm is listed in this section. The Online Tabu Search (TS) algorithm is similar to the TS algorithm listed in the Chapter 2, except for few differences.

Ω : Set of optimized solutions for different standard demands.
S : Current solution.
S^* : Best admissible solution.
$Cost$: Objective function.
$\aleph(S)$: Neighborhood of $S \in \Omega$.
$\mathbf{V}^*$: Sample of neighborhood solutions.
T : Tabu list.
P : The period of time under which the solution has to be found.
AL : Aspiration Level.

Begin
Start with an initial solution $S \in \Omega$.
Initialize tabu lists and aspiration level.
For the time duration P **Do**
    Generate neighbor solutions $\mathbf{V}^* \subset \aleph(S)$.
    Find best $S^* \in \mathbf{V}^*$.
    If move S to S^* is not in **T** **Then**
        Accept move and update best solution.
        Update tabu list and aspiration level.
        Increment iteration number.
    Else
        If $Cost(S^*) <$ **AL Then**
            Accept move and update best solution.
            Update tabu list and aspiration level.
            Increment iteration number.
        EndIf
    EndIf
EndFor
End.

Figure 4.4: Algorithm : Online Tabu Search Algorithm.

The main difference in the two algorithms is the Ω that is used in both the Online TS and the Offline TS algorithm. The Ω of the Offline TS algorithm is the set of random feasible solutions, whereas the Ω of the Online TS algorithm is the set of optimized solutions that were generated for different standard demands. The time required to terminate the algorithms is also different. The Online TS algorithm is allowed to run for a very short duration compared to the Offline TS algorithm. To summarize, the Online TS algorithm is similar to the Offline TS algorithm, but with shorter time duration and a different set of initial solutions.

Chapter 5

Experimental Results for the Online TS

The experimental results for the online simulation processing system implemented by using the TS iterative heuristic are presented here. The results are organized in graphs and tables to give a clear view of the trends.

5.1 Test Cases

Seven different topologies covering three types of graphs *viz.* Hierarchical, Random, and Waxman Graphs were used as test cases in this work. These are depicted in Table 5.1. The testcases and the demand files has Table 5.1. These testcases and the demand files has also been used as input in the offline TS work done by Asad

et al in [25]. For each topology the input files include graph, demand, and capacity. The graph input file contains information regarding nodes and arcs showing connectivity of the entire network. The demand file contains the values of demand (traffic flow) from each source node to every other destination node in the network and the capacity file indicates the capacity of each arc. Table 5.2 shows different levels of demand values for six topologies. All these demands were used in the experiments. The traffic demand that is used as Random traffic demand is called DR. The value of the random demand is 23216 and it is used in all the experiments. The random demand is different from all the standard traffic demands of all the network topologies.

Table 5.1: Test Cases.

Test Code	**Network type**	**N**	**A**
h100N280a	2-level hierarchical graph	100	280
h100N360a	2-level hierarchical graph	100	360
r100N403a	Random graph	100	403
r100N503a	Random graph	100	503
r50N245a	Random graph	50	245
w100N391a	Waxman graph	100	391
w100N476a	Waxman graph	100	476

Table 5.2: Demand Table.

Demand	r100N403a	r100N503a	w100N391a	w100N476a	h100N280a	h100N360a
D4	23099	33531	16158	21164	1535	4136
D6	34648	50297	24237	31747	2303	6203
D8	46198	67063	32316	42329	3070	8271
D10	57747	83829	40395	52911	3838	10339
D11	63522	92211	44434	58202	4221	11373
D12	69297	100594	48474	63493	4605	12407

5.2 Results for Online Processing using Tabu Search

5.2.1 Performance Metrics

Experimental results have been recorded for the following four performance metrics:

1. Cost
2. Maximum Utilization (MU)
3. Percentage of Extra Load (PXLoad)
4. Number of Congested Links (NOCL)

The utilization of the link is the ratio of load on the link to its capacity. If the utilization of the link is more than one, the link is congested. The Maximum Utilization is the utilization of the maximum utilized link in the network. In other words, it is the utilization of the link having the highest degree of congestion. The extra load on a particular link is the load present in excess to its capacity. If the load on the link is less than its capacity ($utilization < 1$), then the extra load on that link is zero. The percentage of extra load is the sum of the extra load present in the network divided by the sum of capacities of congested links. Congested links are the links which have a utilization greater than 1 (i.e., load on the link exceeds its capacity). The statistics for these performance metrics are plotted with respect to the the initial solution weights applied to the DR along with the solution generated

by the OLS system. The cost of applying the Offline TS for 1 hour to the new traffic DR is displayed as well, and is used as a baseline to compare the Online TS with the Offline TS method. The X axis displays the cost, and the Y axis displays the scaled demand.

5.2.2 Cost

Figure 5.1 shows the cost curve for DR in h100n280a when applied with different demand weights as initial solutions. The baseline is the cost achieved when the Offline TS is applied to the new traffic demand DR. The baseline cost will be the best of all because the solution weights are obtained by applying the Offline TS using DR for 1 hour. The mean of the cost if we use the different demand's optimal weight solution as the initial solution and apply it to the new traffic DR is 12 % higher or more than the cost that would be generated by applying the Offline TS for 1 hour. The mean of the cost of applying the Online TS to the random traffic demand DR is 2 % higher than the cost generated by applying the Offline TS for 1 hour to the random traffic demand DR. The application of the Online TS to the initial solution used of a different traffic demand to DR gives a 10% improvement in just 360 seconds. We know that the Offline TS starts with a random initial weight settings to look for an optimal solution. An improvement of 21% of the cost is achieved, if the optimal weight settings of standard demands are used as initial solution, instead of random weight settings. The above improvement is an average

over all test case results.

D4 weight solution is the best initial solution if no further improvements are done. D6 weight settings is the best weight settings solution in terms of cost after applying the Online TS. D10 weight settings shows the best improvement obtained by applying the Online TS, eventhough the initial cost of applying D10 weight settings is the highest.

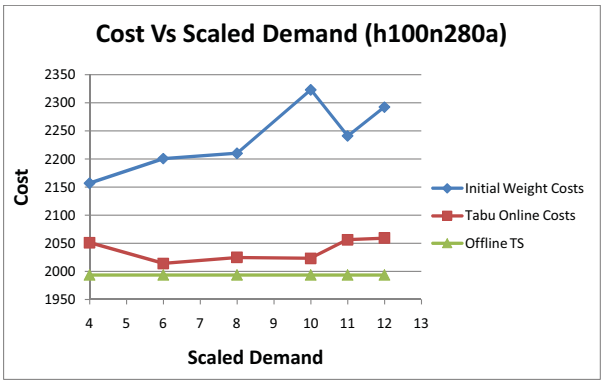

Figure 5.1: Cost Versus Scaled Demand on h100n280a Network .

Table 5.3: Summarized results for all test cases.

Topology	**Measure1=OW/Offline**	**Measure2=Online/Offline**	**Measure3=IW/OW**
r100n403a	1.042	1.017	43
r50n245a	2.82	1.06	0.5
r100n503a	1.078	1.02	11
w100n476a	1.12663	1.039	84
w100n391a	1.1382	1.0503	223
h100n280a	1.12	1.02	1.21
h100n360a	3.04	1.64	3.09

The average of the cost is calculated over all the demands for each network topology,i.e., the average cost of Online TS is calculated by taking the mean of the online costs for D4, D6, and so on. Column 1 of Table 5.3 lists the network topologies used in the simulation. Measure 1 used in column 2 is a ratio, the numerator of the ratio is the average cost achieved when optimal weight solution of a standard demand is used as initial solution, and the denominator is the cost of applying the Offline TS to the DR. Measure 1 is a metric of cost performance of optimal weight settings of a standard demand is used as solution for DR in comparison to the Offline TS cost for DR. Measure 2 used in column 3 is also a ratio, where the numerator of the ratio is the average cost when the Online TS is applied, and the denominator is the cost of applying the Offline TS to the DR. Measure 2 is metric to compare the costs of the Online TS solution compared to the Offline TS solution for DR. Measure 3 in column 4 is a ratio, whose numerator is the cost of the random initial solution used by the Offline TS, and the denominator is the average cost achieved when optimal weight solution of a standard demand is used as the initial solution. Measure 3 is the metric to compare the cost of the random initial solution used by the Offline TS with the cost of the initial solution used by the Online TS. Let us look at the results of r100n503a, the measure 1 indicates that if the optimal weight solution of a standard demand is used as the initial solution, the cost is increased by just 7% compared to the cost generated by the Offline TS which should be the best. Measure 2 indicates that by applying the Online TS, the cost is increased by

just 2% compared to the cost generated by the Offline TS. Measure 2 also indicates a 5% improvement in the cost when compared to Measure 1. Measure 2 is the cost generated by applying Online TS to the initial solution.

Based on the results obtained, it is recommended that the weight solution of a standard demand should be used as an initial solution for finding the optimal weights for the random demand. To further improve the cost, Online TS should be applied to this initial solution. The optimal weight solution of the standard traffic demand used as initial solution is much better in terms of cost when compared to the initial random weight solution used by Offline TS. If the time is not sufficient enough to employ Online TS, using the weight solution of the standard demand as a solution is still highly recommended. If there is enough time, Online TS should be employed to improve the cost further. The results for the best cost for other topologies are shown in Appendix-A.

5.2.3 Maximum Utilization

Taking the same test case h100n280a, we further present the other performance metrics discussed earlier. Higher Maximum Utilization indicates more traffic demand than the link capacity and is bad for the network performance. The Maximum Utilization in the Figure 5.2 is the Maximum Utilization of the standard demand weight settings solution used for DR. Figure 5.2 shows the comparison of the Max-

imum Utilization in the network before and after applying the Online TS. For D11, the maximum utilization has been reduced considerably after applying the Online TS. The results for other demands maximum utilizations are very close to what they were before applying the Online TS, except for D10. The D6 maximum utilization has been increased after applying the Online TS, eventhough the D6 cost has improved. This shows that the improvement of the initial weight solution to reduce the cost could result in an increase of the maximum utilization. To summarize, the maximum utilization of the initial solution is increased after applying the Online TS. In most test cases, maximum utilization results of the Online TS are better when compared to the Offline TS.

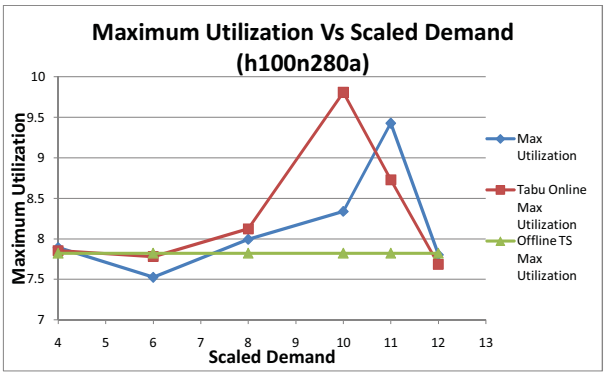

Figure 5.2: Maximum Utilization Versus Scaled Demand on h100n280a.

5.2.4 Number of Congested Links

The next performance metric is the Number of Congested Links (NOCL). The results for the network h100n280a are presented in Figure 5.3. The number of congested links of the Online TS solution for different demands are close to the number of congested links of the Offline TS solution, except for D6. For D11, the number of congested links of the Online TS and the Offline TS solution are same. The number of congested links in the Online TS solution for D6 are less compared to the Offline TS solution. The number of congested links in D6 have increased after applying the Online TS, eventhough D6 cost has improved. This shows that the improvement of the initial weight solution to reduce the cost could result in an increase in the number of congested links. The number of congested links exist if the maximum utilization is higher than 1. For all the test cases that have a maxumum utilization under 1 the number of congested links are zero.

To summarize, the number of congested links of the initial solution are increased after applying the Online TS. The number of congested links in the solutions generated by the Online TS and the Offline TS for the same demand are close.

5.2.5 Percentage of Extra Load

Finally, the results for the Percentage of Extra Load (PXLoad) for the test case h100n280a are shown in Figure 5.4. The extra load is a very good measure of the

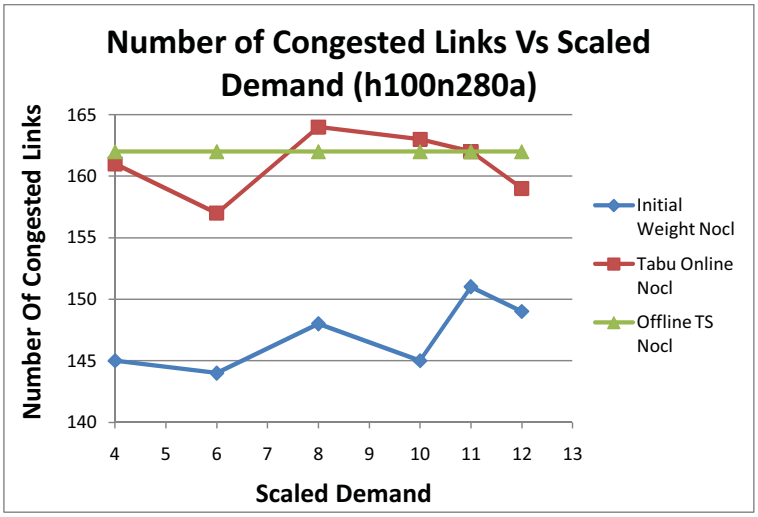

Figure 5.3: Number of Congested Links Versus Scaled Demand on h100n280a Network.

amount of congestion in the network. For D12, the PXLoad of the Offline TS and the Online TS is same. The PXLoad in D6 has decreased after applying the Online TS, along with the cost. This shows that the improvement of the initial weight solution to reduce the cost could result in an improvement in PXLoad. The PXLoad is reduced because of the improvement in the maximum utilization and the reduction in the number of congested links. The PXLoad is zero if the number of congested links are zero, and the maximum utilization is under 1. For all the test cases, the PXLoad is zero if the maximum utilization is under 1 and the number of congested links are zero. However, in few test cases, the PXLoad increases with the increase in the maximum utilization and the number of congested links.

To summarize, PXLoad of the initial solution is reduced after applying the Online

TS. The Offline TS also improves the PXLoad of the initial solution.

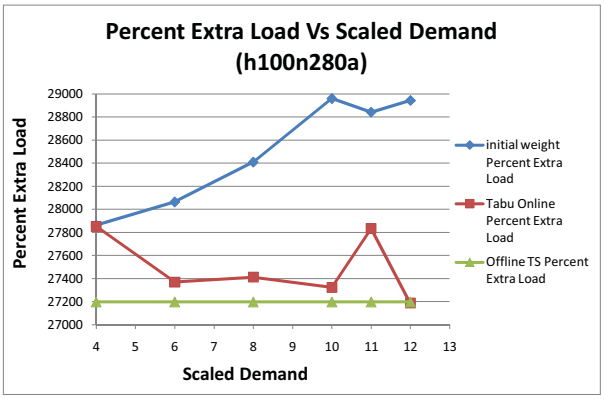

Figure 5.4: Percentage of Extra Load Versus Scaled Demand on h100n280a Network.

The cost results for all the test cases are shown in Appendix-A for reference. The cost comparison shows that the Online TS was found to improve all demand solution weights costs. The most important point of the Online TS was its ability to improve costs even when using higher demands for initial weight solutions.

Higher demands have additional load on the network and hence have high possibility of links getting congested. The solutions for these higher demands use weights that consider this additional load and try to avoid links congestion. To avoid links congestion, higher weights to these links (that have to bear the additional load) are assigned. When this weight is applied to other random traffic demands like DR, it would result in the under utilization of these links (because these links do not have additional load). The concept of using another demand's weight settings as the initial solution to the new traffic demand at hand gives better starting point for

the random traffic demand solution in comparison to the random solution used by the Offline TS.

Chapter 6

Deterministic Approach

6.1 Motivation

The Deterministic Approach (DA) is an attempt at finding the solution for the OSPFWS online problem in a deterministic way. The DA takes into consideration that the weight solution that is being used as the initial point is from a traffic demand which is different from the one for which a solution is being searched. The DA looks at both the demands, i.e., demand used for the initial solution and the given traffic demand. Basically, the DA looks at the difference in both the demands. To understand how the DA works, let us look at it step by step. Let us say that we have a random traffic demand called DR for a network. To search for the optimal solution weights for this traffic demand, using an optimal weight solution of a different standard traffic demand say D8 as an initial solution is better than

using a random weight solution was already shown in Chapter 5 . The experimental results discussed in the Chapter 5 show that applying a different weight solution of a standard demand as the initial weight setting gives an improved and better cost in comparison to the random initial solution used by the Offline TS. We are using the D8 weight solution as the initial weight setting for DR. After the application of D8 weights, the calculation of the cost is done. Then, the process of refining the weights that have been applied is started. The aim of the refinement process is to improve on this cost to lead to a better solution resulting in better performance of the network. So, if we observe that DR is different from D8 and hence applying D8 weight solution for DR does not guarantee the best solution for DR. Then, this D8 weight solution has to be refined for DR. If the network topology is the same, the capacities of the links remain the same, and the only changed input is the demand, then it is safe to say that the demand has an influence on the weights that are being selected for the optimal solution. We have been able to say in the above lines that if a deterministic approach has to be built, then it should exploit the influence of the input traffic demand on the weight solution. The DA does exactly that, it takes the initial solution as the weights solution of D8 and applies them to DR. The outgoing demands from each node in D8 are added up. The outgoing demands of each node in DR are also added. The DA then finds the nodes which have different traffic demands in D8 and DR. DR and D8 ratio is calculated to get the difference in the demands at the node level. From all these nodes, the one with the highest ratio is

selected. The highest ratio is selected because it represents the demand at a node in DR which is the most different from the D8. Hence, the weights that are being used at this node needs to be changed. In order to suit the demand in DR, the weights at all the outgoing links at this node are multiplied with the ratio that has been calculated earlier. The approach uses only the outgoing demands for comparison and refinement. The DA refines the weights of all the outgoing links of this single node, which has the most different traffic demand in DR compared to the D8. The following section details the algorithm of the DA approach.

6.2 Deterministic Approach(DA) Algorithm

M	is the number of outgoing links from j.
N	is the number of nodes in the network.
$p[j][K]$	is the traffic demand from node j to node K in demand initial.
$d[j][K]$	is the traffic demand from node j to node K in DR.
$W[j][K]$	is the weight at the outgoing link from node j to node K
$NW[j][K]$	is the new weight at the outgoing link from node j to node K

START

1. Receive input network graph.

2. Receive input network resource capacity and store capacity for cost and utilization calculation of the simulated network.

3. Receive input new network traffic demand and store traffic demand DR.

4. Take the nearest demand weight solutions (say demand initial) to the given DR and then apply these weights to the DR. (Here we are using the weight solution of the demand that is closer to our DR, as the initial solution).

5. Compute the sum of the demands of all outgoing links at each node in the initial demand as:

   For $j = 1$ to N **Do**

   For $K = 1$ to M **Do**

   $di[j] = di[j] + p[j][K]$;

   EndFor K

   EndFor j

6. Compute the sum of the demands of all outgoing links at each node in DR as

   For $j = 1$ to N **Do**

   For $K = 1$ to M **Do**

   $dr[j] = dr[j] + d[j][k]$;

   EndFor K

   EndFor j

7. Compute and maintain an array 'Ratio' for all the ratios, for each node of the network.

   For $j = 1$ to N **Do**

   Ratio[j]=dr[j]/di[j];

   EndFor j

8. Find the largest value in array 'Ratio' called dL.

   Set dL=0 and index=0;

   For $j = 1$ to N **Do**

   If $dL < Ratio[j]$ **Then**

   Set dL=Ratio[j];

   Set index=j;

   EndIf

   EndFor j

9. Maintain an array 'XY' and store the demands at all outgoing links from the node that corresponds to the ratio dL. (getting all the demands whose weights has to be recomputed)

   Set $j = index$

   For $K = 1$ to M **Do**

   $XY[K] = d[j][K]$;

   EndFor K

10. Compute new weights 'NW'.

    Set $j = index$

    For $K = 1$ to M **Do**

    $NW[j][K] = W[j][K] * dL$;

    EndFor K

11. Assign the new weights to the outgoing links of the node that correspond to the ratio dL.

    Set $j = index$

    For $K = 1$ to M **Do**

    $W[j][K] = NW[j][K]$;

    EndFor K

12. Calculate the cost;

End

Figure 6.1: Algorithm : Deterministic Approach Algorithm.

Chapter 7

Experimental Results for Online Processing using Deterministic Approach

7.1 Results for Online Processing using DA

7.1.1 Cost

Figure 7.1 shows the cost curve for DR in h100n280a when applied with different demand weight solutions as initial solution. The baseline is the cost achieved when the Offline TS is applied to the new traffic demand DR. The baseline cost will be the best of all cases because the solution weights are searched by applying TS for

1 hour. The best weight setting to be used as an initial solution is the one with the least cost and in the Figure 7.1 it is D4. The D4 weight setting solution gives the best cost when compared to the rest of the demands before applying the DA. The D4 weight setting solution gives the best cost when compared to other demands even after applying the DA. The D10 weight solution is the best improvement after applying DA. Eventhough D10 has the best improvement, the solution cost is high compared to the other demands. The mean of cost of DA improved solution is 11% higher than the cost of the dedicated Offline TS cost. The mean of cost of the initial solution is 12% higher than the cost of the dedicated Offline TS cost. There is a 1% improvement in the cost from the initial solution used to the DA improved solution weights for the same initial solution used.

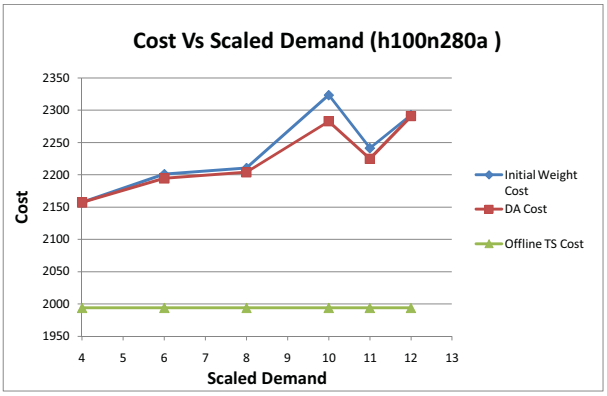

Figure 7.1: Cost Versus Scaled Demand on h100n280a Network .

The cost curves for another test case are shown in Figure 7.2. The best weight

settings to be used as an initial solution is the one with the least cost and in this Figure 7.2 it is D10.

The D10 weight setting solution gives the best cost even after applying the DA and the weights improved by DA. The D11 weight solution has the best improvement in the cost after applying DA. Eventhough D11 weight solution cost has improved, it is still high when compared to the other demands. The mean of cost of DA improved solution is twice better than the cost of the dedicated Offline TS cost. The mean of cost of the initial solution is 5% better than the cost of the dedicated Offline TS cost. There is a 56% improvement of the initial solution cost after applying the DA. The mean of the initial solution is 2.8 times better than the cost of the initial solution used by the Offline TS. To summarize, the DA improves the cost of the initial solution.

Based on the results obtained, it is recommended that the weight solution of a standard demand should be used as an initial solution for finding the optimal weights for the random demand. To further improve the cost, DA should be applied to the initial solution. The optimal weight solution of the standard traffic demand used as initial solution is much better in terms of cost when compared to the Offline TS solution. The weight solution of the standard demand as solution is better than the solution generated by the Offline TS and is recommended. The DA improves the cost of the initial solution and applying DA to the initial solution is highly recommended.

The results for the best cost for other topologies are shown in Appendix-B. The curves indicate that a small gain is achieved for every demands initial solution by applying the DA.

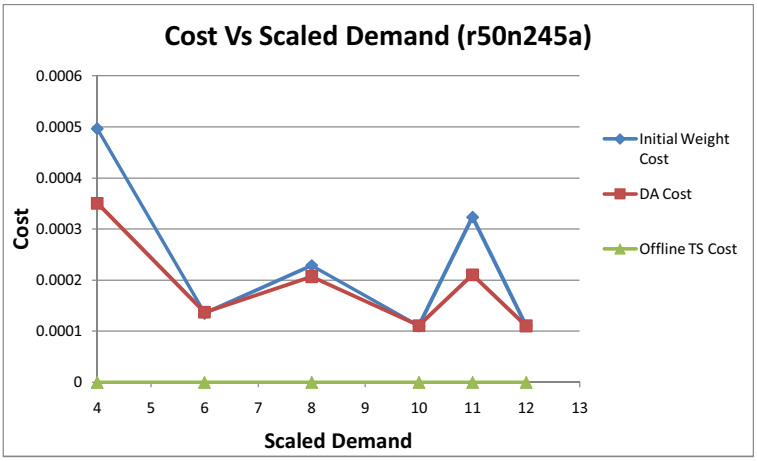

Figure 7.2: Cost Versus Scaled Demand on r50n245a Network.

7.1.2 Maximum Utilization

Taking the same test case h100n280a, we further present the other performance metrics discussed earlier. Figure 7.3 shows the comparison of Maximum Utilization in the network before and after applying DA. Figure 7.3 also shows the Maximum Utilization in the network when Offline TS solution is used. For D11, the maximum utilization has been reduced considerably after applying DA. The results for other demands maximum utilizations are very close to what they were before DA is applied. The D6 maximum utilization decreased after applying DA, and the D10

cost improved after DA processing. The two previous facts could imply that the improvement of the initial weight solution to reduce the cost leads to a decrease in maximum utilization. The decrease of maximum utilization and reduction of cost implies that the links that were previously not utilized properly because of higher weights have been utilized to their proper potential without causing overutilization. To summarize, maximum utilization of the initial solution is reduced after applying the DA.

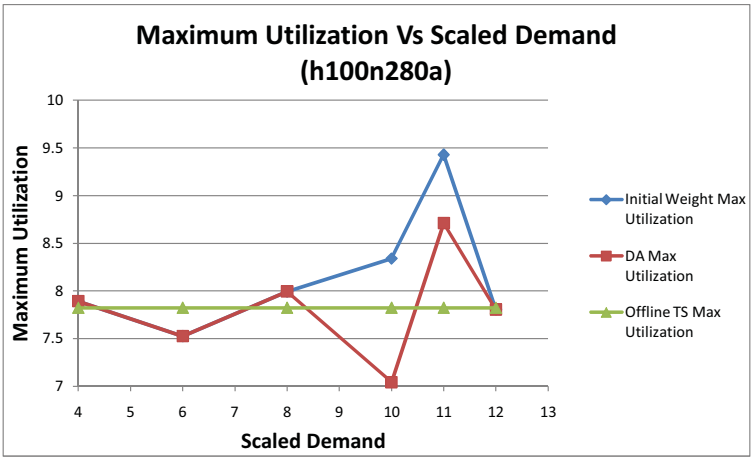

Figure 7.3: Maximum Utilization Versus Scaled Demand on h100n280a.

7.1.3 Number of Congested Links

The next performance metric is the Number of Congested Links (NOCL). The results for the network h100n280a are presented in Figure 7.4. If we look at the Figure 7.4,

we observe that in the process of improving the costs, the number of congested links are increasing. The same behavior has been observed in the Online TS approach results trend. In the D10 solution, the number of congested links has increased from before. To summarize, number of congested links of the initial solution increase after applying the DA.

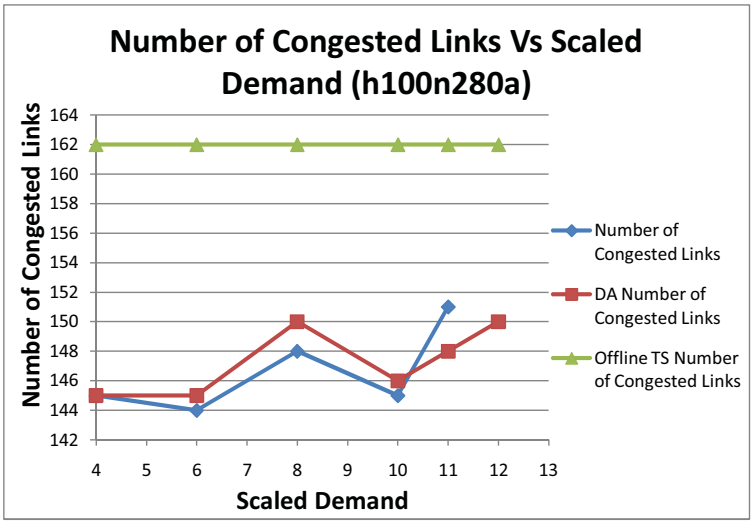

Figure 7.4: Number of Congested Links Versus Scaled Demand on h100n280a Network.

7.1.4 Percentage of Extra Load

Finally, the results for the Percentage of Extra Load (PXLoad) for the test case h100n280a are shown in Figure 7.5. The extra load is a very good measure of the amount of congestion in the network. Results show that the PXLoad of solutions

using higher demands as an initial solution are high when compared to the solutions using lower demands. To summarize, PXLoad of the initial solution is reduced after applying the DA.

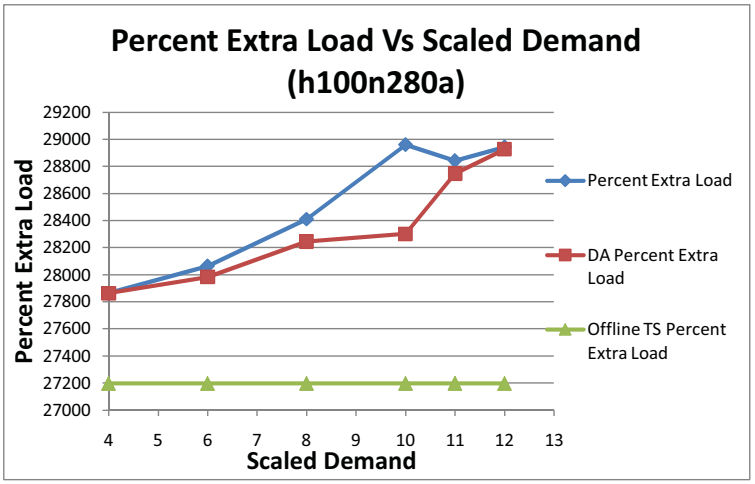

Figure 7.5: Percentage of Extra Load Versus Scaled Demand on h100n280a Network.

Based on the results obtained, it is recommended that, the weight solution of the standard demand should be used as an initial solution for finding the optimal weights for the random demand. The DA should be applied to the initial solution to improve the cost. The DA is the best approach if the time is very short.

7.2 Multiple levels DA

If the initial solution is taken from a traffic demand, e.g., Demand A, to find better weight solutions for another traffic demand DR; then the Ratio matrix has the values that correspond to the ratio of these two traffic demands. The elements in the Ratio matrix are calculated as $Ratio[i] = DemandR[i]/DemandA[i]$. The matrix Ratio is then sorted in a descending order with the largest ratio element at the top. DA is applied by changing the weight of only those links corresponding to the highest ratio between the initial solution traffic demand and the new traffic demand. Changing the weights of the links of the top 2 highest ratios is called as a level 2 DA. The level 2 DA changes the weights of the links corresponding to the highest ratio and the next highest ratio. The level 3 DA includes changes in the weights of the corresponding links of the highest ratio, the second highest ratio, and the third highest ratio. To generalize, an level N DA includes all the weight changes corresponding to the top N ratios. Table 7.1 has the cost results for applying the Multiple levels DA for the network topology h100n280a. Column 1 of Table 7.1 lists the traffic demand matrix which is used as the initial solution. Column 2 contains the cost of using the weights solution of the demand matrix of column 1 as the initial solution for solving the problem for DR. Column 3 of Table 7.1 has the cost of applying level 1 DA (Which is the standard DA). Column 4, Column 5, and Column 6 have the cost of applying the level 2, level 3, and level 4 DA respectively.

Table 7.1: Multiple level DA costs for h100n280a.

Initial solution	**Initial solution cost**	**Level 1**	**Level 2**	**Level 3**	**Level 4**
D4	2157	2157	2308	2325	2390
D6	2201	2194	2350	2351	2372
D8	2210	2203	2334	2349	2355
D10	2323	2282	2483	2484	2527
D11	2241	2224	2410	2412	2497
D12	2292	2290	2427	2442	2590

Table 7.2 includes the cost results for applying the Multiple level DA for the network topology r100n503a.

Table 7.2: Multiple level DA costs for r100n503a.

Initial solution	**solution cost**	**Level 1**	**Level 2**	**Level 3**	**Level 4**
D4	1.08927	1.08566	1.11742	1.1433	1.16118
D6	1.08165	1.08339	1.10654	1.1339	1.14774
D8	1.17103	1.16728	1.21417	1.25054	1.24966
D10	1.11321	1.11091	1.13513	1.16824	1.17289
D11	1.15266	1.15079	1.20606	1.2394	1.23681
D12	1.10617	1.1074	1.13096	1.15831	1.16964

If we look at the costs generated by applying the multiple level DA on both networks listed in Table 7.1 and Table 7.2, we observe that the multiple level DA does not generate better costs. The more levels we use the worse the results obtained, indicating that multiple level DA does not yield better results in comparison with the standard DA.

7.3 Node Level DA

In section 7.2, we have seen the working of the multiple level DA and have found that the results are not encouraging to proceed to further levels. Another direction in DA is also probed with an intent to improve the cost of the solution generated by the DA. Node level DA is similar to the multiple level DA with the difference being the exclusivity of changing weights of only a single corresponding node. To implement a node 2 level DA the weights of the links corresponding to the second highest element of the Ratio matrix are altered, without changing the weights corresponding to the highest element of the Ratio matrix. Earlier in the Multiple level DA, an N level DA would change the weights of all the links corresponding to the top N elements of the Ratio matrix, however in the Node level DA only the weights of the corresponding links of the N^{th} node are changed. Table 7.3 lists the cost results for applying the Node level DA for the network topology h100n280a.

Column 1 of Table 7.3 lists the traffic demand matrix which is used as the initial solution. Column 2 contains the cost of using the weights solution of the demand matrix of column 1 as the initial solution for solving the DR problem. Column 3 of Table 7.3 has the cost of applying node 1 level DA (Which is the standard DA). Column 4, Column 5, and Column 6 have the cost of applying the node 2 level, node 3 level, and node 4 level DA respectively.

Table 7.3: Node level DA costs for h100n280a.

Initial solution	**solution cost**	**Node 1**	**Node 2**	**Node 3**	**Node 4**
D4	2157	2157	2168	2234	2345
D6	2201	2194	2200	2224	2342
D8	2210	2203	2199	2246	2340
D10	2323	2282	2324	2347	2437
D11	2241	2224	2254	2370	2396
D12	2292	2290	2249	2373	2446

Table 7.3 lists the cost results for applying the Node level DA for the network topology h100n280a. Table 7.4 lists the cost results for applying the Node level DA for the network topology r100n503a.

Table 7.4: Node level DA costs for r100n503a.

Initial solution	**solution cost**	**Node 1**	**Node 2**	**Node 3**	**Node 4**
D4	1.08927	1.08566	1.11322	1.10665	1.08488
D6	1.08165	1.08339	1.10936	1.09917	1.0856
D8	1.17103	1.16728	1.19962	1.17216	1.17141
D10	1.11321	1.11091	1.13797	1.12088	1.11485
D11	1.15266	1.15079	1.18286	1.15007	1.15568
D12	1.10617	1.1074	1.12982	1.11967	1.11223

If we look at the results of the cost generated by applying the Node level DA in comparison with the multiple level DA, Node level DA is better. However, if we look at the trend of improvement in the costs with respect to the node levels, there is no relation as such that node i level results are better than node j level. For D4, the Node 4 level DA cost is better than the Node 1 level DA. The results of Node level DA allow us to understand that unlike the multiple level DA, there are few improvements. The less improvement could be because few weight changes are

favorable towards a better solution rather than more weight changes.

7.4 Random Node Level DA

After exploring the Node level DA, another variation of this approach is conceptualized. The Node level DA does the changes to a particular node depending on the elements position in the Ratio matrix. If instead of selecting node number 4 or node number 2 to change the weights, a random node is selected from the Ratio matrix and the corresponding node weights are changed. The results are taken for three demands of two different network topologies. There are four runs for each traffic demand for the random node level DA. Table 7.5 shows the cost results for applying the Node level DA for the network topologies r100n503a and h100n280a for traffic demands D12 and D8.

Table 7.5: Random Node level DA costs.

Network Topology	**Initial solution used**	**Node Random Cost**
r100n503a	D12	1.12209
r100n503a	D12	1.12219
r100n503a	D12	1.12275
r100n503a	D12	1.12211
h100n280a	D12	2288
h100n280a	D12	2296
h100n280a	D12	2290
h100n280a	D12	2314
h100n280a	D8	2208
h100n280a	D8	2201
h100n280a	D8	2205
h100n280a	D8	2195

The Random Node level DA cost results are not encouraging and are not better than the Node level DA or the DA.

7.5 Comparison of Online TS and DA

The Online TS gives good results with solution weights that result in lesser cost than the initial solution cost. The weights used as the initial solution are improved by applying the Online TS and the resultant weights solution gives better cost. The gain in cost by applying the Online TS is good. If we compare the Online TS and DA, the Online TS obtains weight solutions which result in better cost than the DA. The Online TS searches for new weights starting from the initial solution weights looking for a better cost in the neighboring space of the initial solution. DA takes very little time in finding the new weights which results in a better cost than the initial weight solution cost. Eventhough the Online TS is able to find a better cost at the end, the cost of the initial solution improves slowly at the start. DA starts improving the cost immediately from the initial solution with very few changes from the initial solution weights. The time taken by the DA and the Online TS to come up with weight settings for a better cost is different. DA takes 5 seconds to come up with the new weight settings that improves the cost from the initial weight setting solution cost. The Online TS is used for 360 seconds to search for the new weight settings which results in an improved cost from the initial weight setting solution

cost. To compare the performance of Online TS and DA in finding weights that result in a better cost, the allowed time for both must be the same. The time taken by the Online TS is reduced to the time taken by DA to find the new weights. In other words, the Online TS is applied for 5 seconds to search for solution weights that result in an improved cost starting from the cost of applying the initial solution weights. The comparison is done for two network topologies, namely h100n280a and r100n403a. The traffic demands that are used for the initial solution in both the network topologies are D4, D6, D8, D10, D11, and D12. Figure 7.6 shows the cost curves for the Online TS and the DA applied for 5 seconds for the network topology h100n280a.

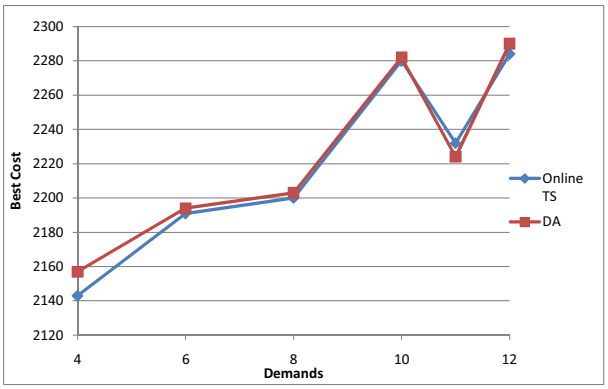

Figure 7.6: Comparison of Online TS Cost and DA Cost Versus Scaled Demand on h100n280a Network .

The cost generated by the Online TS is achieved by finding the new weights starting from the initial solution and applying Online TS for 5 seconds only. Figure 7.7 shows the cost curves for Online TS and DA applied for 5 seconds for the network

topology r100n403a. Figure 7.6 and Figure 7.7 show that eventhough Online TS is applied for 5 seconds, the cost generated is very near to the cost generated by DA. Another point worth noting again is that the Online TS changes many weights in each step, while DA on the other hand just changes few weights. The observation based on the results of DA and Online TS cost values is that the DA gives better or similar cost to the Online TS if the time is limited to 5 seconds.

To summarize, if the time limit is 5 seconds, then DA is better in finding the best cost weight settings. If the time limit can be expanded and is around 360 seconds, then the Online TS obtains the weight settings which when employed gives much better cost than the cost of applying the initial weight settings solution.

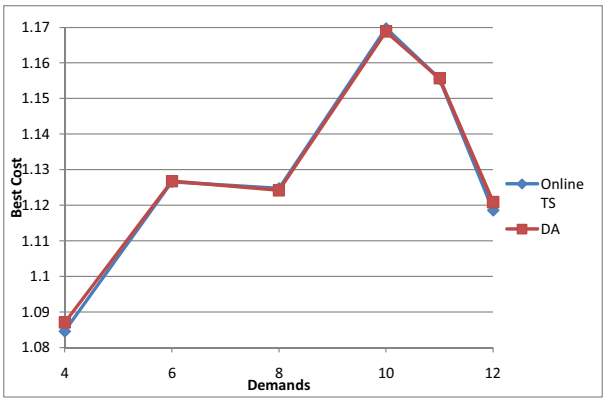

Figure 7.7: Comparison of Online TS cost and DA Cost Versus Scaled Demand on r100n403a Network .

7.6 Comparison of OLS and Node level DA

In the earlier sections, we have disccussed the Node level DA. The comparison that is being done here is between the Online TS and the best costs of the Node level DA. In this section, we are going to compare the performance of the Online TS and the best costs of node level DA in finding a better cost solution weights in 30 seconds. The concept of best costs of node level DA is explained as follows. We start with the weight setting of another demand as the initial solution for a given demand at hand, and then apply DA to the weight resulting in new weights which when applied give a better cost. The process described above is called DA, level 1 DA, or node 1 level DA. The time spent in calling DA is 5 seconds. Calling node 2 level DA on the same problem with the same initial solution takes 10 seconds. Calling node 3 level DA on the same problem with the same initial solution takes 15 seconds. The time increases by 5 seconds with the increase in the levels of the node level DA. The cost of node level DA applied for 5 seconds is the cost achieved by the node 1 level DA. The cost of node 2 level DA is the best of the costs achieved by the DA and the node 2 level DA. Similarly, the cost of node 3 level DA is the best of the costs achieved by the DA, the node 2 level DA, and the node 3 level DA. Similarly with the time increase, the best of the costs achieved by the DA and the all the node n level DA applied within that time duration is used. The selection of the best of all the costs is done to include the better costs achieved by higher node level DA. The

comparison gives us a picture of how the node level DA behaves in the backdrop of Online TS direction of finding the best cost. The comparison is done of best cost versus time of both the node level DA and the Online TS.

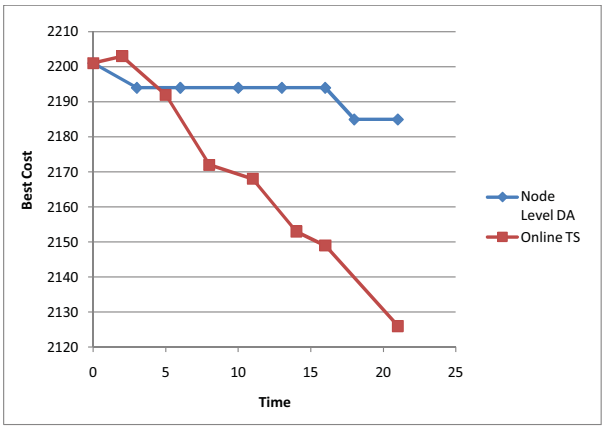

Figure 7.8: Comparison of Online TS cost and node level DA Cost Versus time for D6 on h100n280a Network.

Figure 7.8 is the cost comparison for traffic demand DR using the D6 as the initial solution for the network topology h100n280a using both the Online TS and the best costs of Node level DA. Both the Online TS and node level DA starts with the same initial cost. Node level DA obtains a better cost than the Online TS under 5 seconds, but after that, very good cost improvements are seen in the Online TS whereas the performance of the node level DA does not improve much. An improvement in node level DA cost is seen at the 6 level DA.

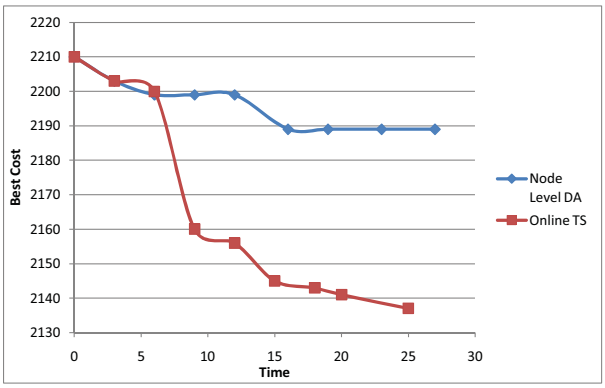

Figure 7.9: Comparison of Online TS cost and node level DA Cost Versus time for D8 on h100n280a Network.

Figure 7.9 is the cost comparison for traffic demand DR using the D8 as the initial solution for the network topology h100n280a using both the Online TS and the best costs of Node level DA. The Online TS and the DA have the same costs for the initial solution and for the solution at time 3 seconds. The cost of node level DA solution at time 6 seconds is a little better than the Online TS solution. The cost of the solution generated by the Online TS after the time 6 seconds is much better than the cost of the solution obtained by the DA. The performance of DA remains the same, and an improvement is seen only after the time period of 15 seconds.

Figure 7.10 is the cost comparison for the traffic demand DR using the D10 as the initial solution for the network topology r100n503a using both the Online TS and the best costs of Node level DA. The Node level DA obtains a considerable better cost when compared with the Online TS at time 6 seconds. At time 10 seconds,

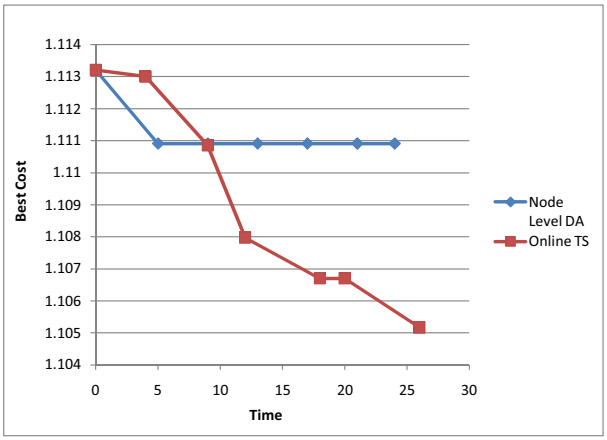

Figure 7.10: Comparison of Online TS cost and node level DA Cost Versus time for D10 on r100n503a Network.

the Online TS and the Node level DA has the same cost. The performance of Node level DA remains the same throughout. The Online TS is able to reach a better cost weight setting solution in the end.

Chapter 8

Conclusion and Future Work

This Final chapter summarizes the thesis work and its contributions to the OSPFWS online simulation system. Section 8.1 provides the conclusions drawn from the thesis work. Section 8.1 also highlights the main contributions of the thesis. Future research based on this thesis and general directions in related areas are described in section 8.2.

8.1 Conclusion

This research work focused on adapting a modern iterative heuristic to solve the OSPF Weight Setting problem for the online system. The problem was addressed by implementing Tabu Search in finding the solutions for the online system. TS was proved to be an elegant heuristic which improved the initial weight solution

of a different demand for the new changed traffic demand. It was also shown that the DA can be applied to refine the initial weights towards an optimal solution for the given traffic demand. The TS online system gave a very good improvement of the initial weight setting of a different traffic demand, and resulted in an optimal solution weights for the new traffic demand. TS was used because it provides a very good solution (Cost) for large topologies at highest demands. The Deterministic Approach improvement on the initial results was limited, however the time spent in generating the new results is limited as well. Eventhough the OSPFWS problem is NP-hard, DA was successful in improving the initial weights of the traffic demand and improve the costs for applying these weight solutions on the given traffic demand. The offline system used a random solution to start the search for the solution, but the online system uses near or approximate demands weight solution as the initial solution. Both the approaches has proved that weight solutions that have been found optimal for a particular traffic demand can be used as initial weight solutions for a different traffic demand and reach a better solution. Different runs of the DA with different formulas have also emphasized that there is no direct relation between the demands and their weights, i.e., a demands solution weights cannot be increased or decreased to suit a different demand which is higher or lower respectively, indicating the complexity of the multiconstrained problem. The improvement of the DA is little because the change has been made to very few weights.

The main contributions of this thesis work is the design and implementation of the

online simulation system using TS. The idea of using the initial solution of standard traffic demands for searching the solution space for the given traffic demand is also a contribution. The formulation of the DA in finding the solution for the traffic demand at hand is a major contribution.

Extensive work has been done in the area of the Deterministic Approach. Weight changes done with multiple levels with the DA approach do not yield better results. The change that has been done in the random fashion for the link weight depending on the difference between the two traffic demands does not result in a good cost reduction. Exclusive node link weight change based on the difference between the two traffic demands is better than the multiple level cost improvement.

8.2 Future Work

The online system can be further refined by applying different iterative heuristics or some hybrid models which better suit the time constraint. The DA could be improved further if the number of weight refinements are increased and as well as exploring different ways for the refinement of the weights other than the one used here. The Deterministic Approach could also be used for problems which are similar and look for solutions starting at an initial solution. The Deterministic approach can further be improved by working on the individual difference of demands between the initial and the new demands. The online processing system using Online TS and

DA can be further developed into a complete software package by integrating them with a front end graphical user interface. A different direction could also be explored by implementing the OLS modules monitor and configure on a testbed.

Bibliography

[1] Neilsen. World internet usage statistic news and world population stats. World Wide Web electronic publication, 2009.

[2] Internet Engineering Task Force. Overview and principles of internet traffic engineering. *Informational RFC 3272*, 2002.

[3] Richard A. Cawley. *Interconnection, pricing, and settlements: some healthy jostling in the growth of the Internet*. MIT Press, 1997.

[4] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Prentice Hall Series, 2002.

[5] Mani Subramanian. *Network Management, principles and practice*. Addison Wesley, 2000.

[6] Internet Engineering Task Force. OSPF version 2. *Technical Report RFC 1583*, 1994.

[7] cisco. *Cisco handbook of Internet working Technology*. Cisco Systems Inc., 2001.

[8] Hauser and carl. *Internet Routing in computer communication Network Protocols*. Washington State University, 2007.

[9] Richard Mortier. Internet traffic engineering. *Technical Report 532 Cambridge University*, 2002.

[10] Xipeng X, Alan H, Brook B, and Lionel M.Ni. Traffic engineering with mpls in the internet. *IEEE Network Magazine*, 2000.

[11] Internet Engineering Task Force. OSPF version 2. *Technical Report RFC 2328*, 1998.

[12] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice Hall Series, 1992.

[13] A. Khanna and J. Zinky. The revised arpanet routing metric. *ACM SIGCOMM Symposion on Communications Architectures and Protocols*, 1989.

[14] Internet Engineering Task Force. OSPF version 2. *Technical Report RFC 2328*, 1998.

[15] cisco. *OSPF Technical Router Working Guide*. Cisco Systems Inc., 2000.

[16] Caldwell, D. Gilbert, A. Gottlieb, J. Greenberg, A. Hjalmtysson, and G. Rexford. The cutting edge of ip router configuration. *COMPUTER COMMUNICATION REVIEW*, 2004.

[17] Internet Engineering Task Force. OSPF version 2. *Technical Report RFC 1245*, 1991.

[18] Sadiq M. Sait and Habib Youssef. *VLSI Physical Design Automation: Theory and practice*. McGraw-Hill Book Company, Europe, December 1995.

[19] Sadiq M. Sait and Habib Youssef. *Iterative Computer Algorithms and their Application to Engineering*. IEEE Computer Society Press, December 1999.

[20] Y. Seok, Y. Lee, Y. Choi, and C. kim. Explicit multicast routing algorithms for constrained traffic engineering. *Seventh Internation Symposium on Computers and Communications*, 2002.

[21] B. Fortz, J. Rexford, and M. Thorup. Traffic engineering with traditional IP routing protocols. *IEEE Communicatoins Magazine*, pages 118–124, 2002.

[22] Mohammed H. Sqalli, Sadiq M. Sait, and Aijaz M. Mohiuddin. An enhanced estimator to multi-objective OSPF weight setting problem. *Network Operations and Management Symposium, NOMS*, 2006.

[23] Nor Musliza Mustafa and Mohamed Othman. A review of routing optimization using ospf. *IEEE International conference on Telecommunicatins*, 2007.

[24] Mohammed H. Sqalli, Sadiq M. Sait, and Aijaz M. Mohiuddin. Engineering evolutionary algorithm to solve multi-objective OSPF weight setting problem. *Advances in Artificial Intelligence*, 2006.

[25] SYED ASADULLAH. Optimization of metrics in ospf/is-is routing using accelerated iterative heuristics. *Master thesis, KFUPM*, 2007.

[26] Bernard Fortz and Mikkel Thorup. Increasing internet capacity using local search. *Technical Report IS-MG*, 2000.

[27] Bernard Fortz and Mikkel Thorup. Internet traffic engineering by optimizing OSPF weights. *IEEE INFOCOM*, 2000.

[28] G. Ivan, Z. Thomas, R. Fabio, and R. Peter. Adaptive multipath routing for dynamic traffic engineering. *GLOBECOM*, 2003.

[29] A. Viswanathan E. Rosen and R. Callon. Multiprotocol label switching architecture. *IETF RFC 3031*, 2001.

[30] H. Kochkar, T. Ikenaga, K. Kawahara, and Y. Oie. Multiclass qos routing strategies based on the network state. *IEEE Computer Communicatoins Magazine*, pages 1348–1355, 2005.

[31] C. Villamizar. Ospf optimized multipath (ospf-omp). *IETF Internet Draft*, 1999.

[32] Hitomi Tamura, Tsuyoshi Okubo, Yousuke Inoue, Kenji Kawahara, and Yuji Oie. Implementation and experimental evaluation of on-line simulation server for ospf-te. *Seventh International Conference on Hybrid Intelligent Systems*, 2007.

[33] Hema Tahilramani Kaur, Tao Ye, Shivkumar Kalyanaraman, and Kenneth S. Vastola. Minimizing packet loss by optimizing ospf weights using online simulation. *Proceedings of the 11TH IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunication Systems*, 2003.

[34] T. M. Thomas II. *OSPF network design solutions.* Cisco Press., 1998.

[35] T.ye, D.Harrison, B.Mo, B.Sikdar, H.T.Kaur, S.Kalyanaraman, B.Szymanski, and K.Vastola. Network management and control using collaborative on-line simulation. *IEEE ICC Proceedings*, 2001.

Appendix-A

A.1 Online TS Results

A.1.1 r50n245a

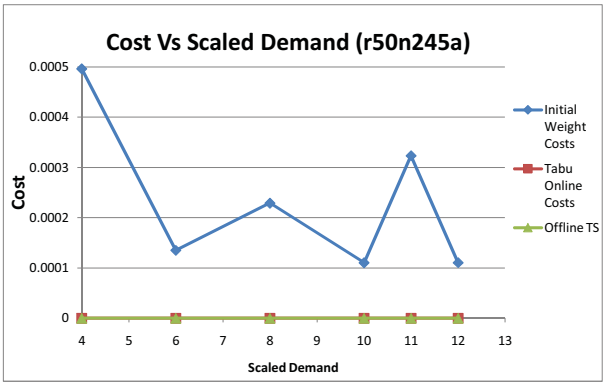

Figure A.1: Cost Versus scaled demand on r50n245a network .

A.1.2 w100n391a

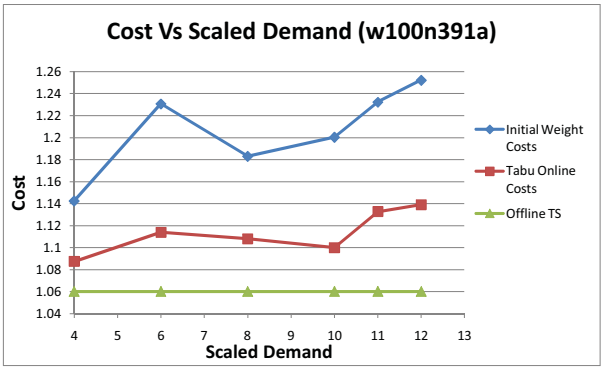

Figure A.2: Cost Versus scaled demand on w100n391a network.

A.1.3 w100n476a

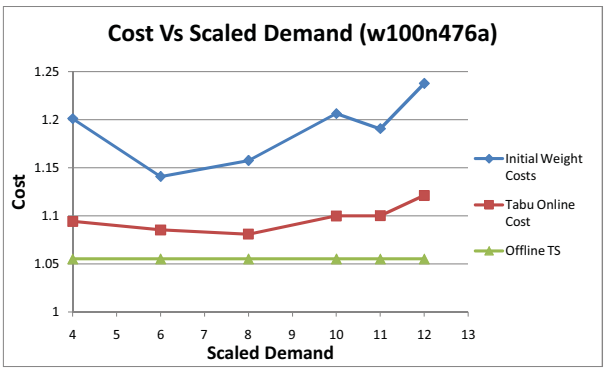

Figure A.3: Cost Versus scaled demand on w100N476a network.

A.1.4 h100n280a

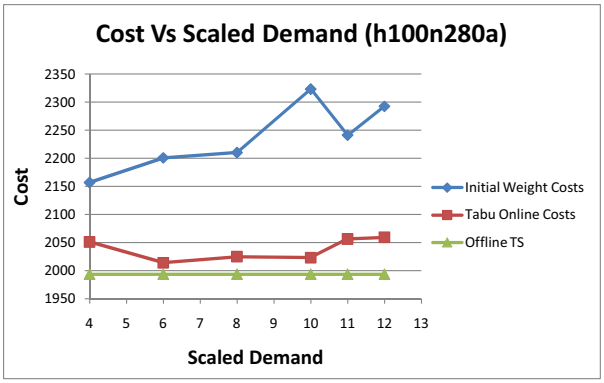

Figure A.4: Cost Versus scaled demand on h100n280a network.

A.1.5 h100n360a

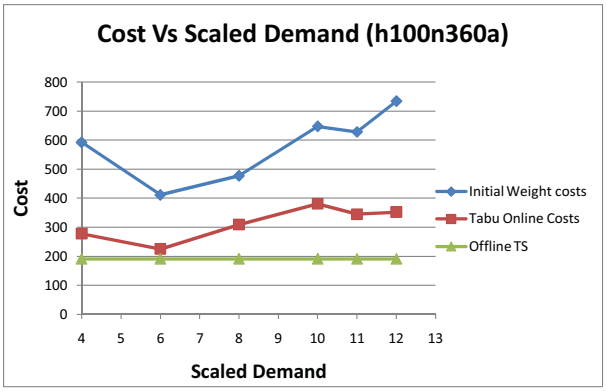

Figure A.5: Cost Versus scaled demand on h100n360a network.

A.1.6 r100n403a

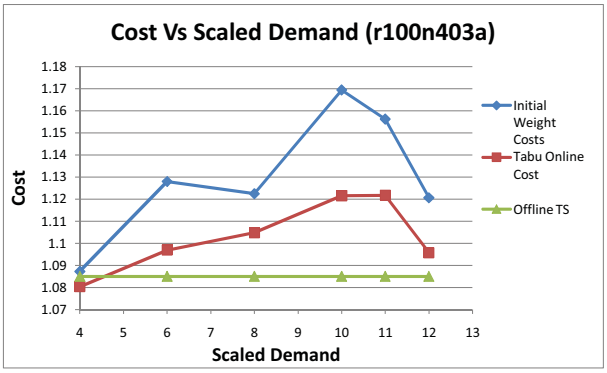

Figure A.6: Cost Versus scaled demand on r100n403a network.

A.1.7 r100n503a

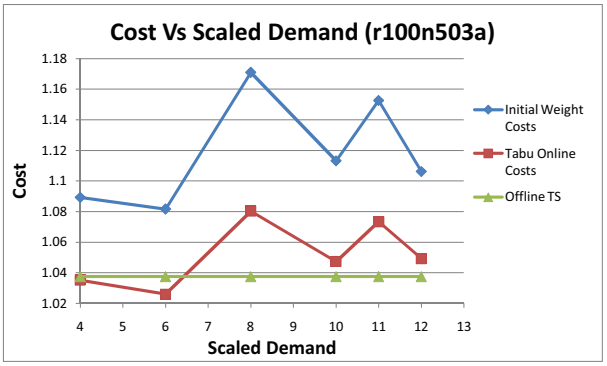

Figure A.7: Cost Versus scaled demand on r100n503a network.

Appendix-B

B.1 DA Results

B.1.1 r50n245a

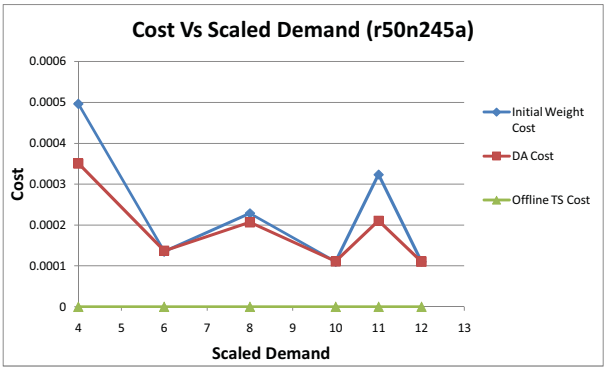

Figure B.1: Cost Versus scaled demand on r50n245a network .

B.1.2 w100n391a

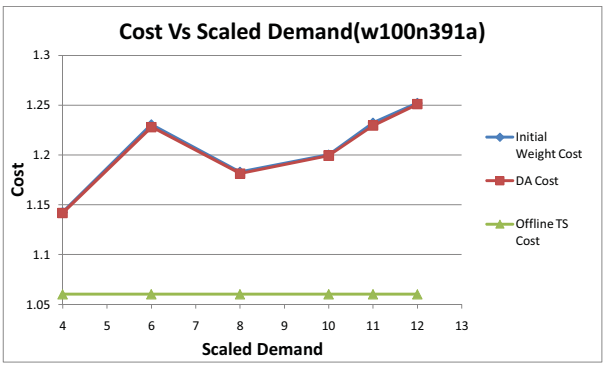

Figure B.2: Cost Versus scaled demand on w100n391a network.

B.1.3 w100n476a

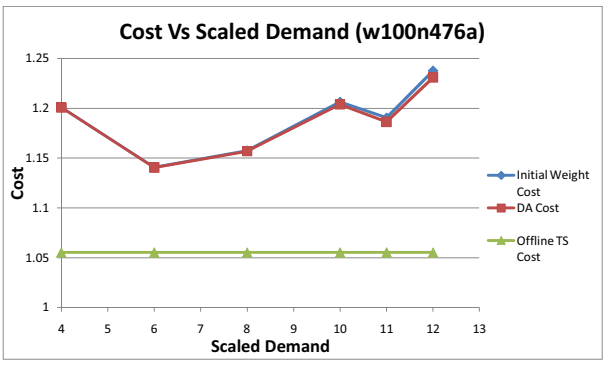

Figure B.3: Cost Versus scaled demand on w100N476a network.

B.1.4 h100n280a

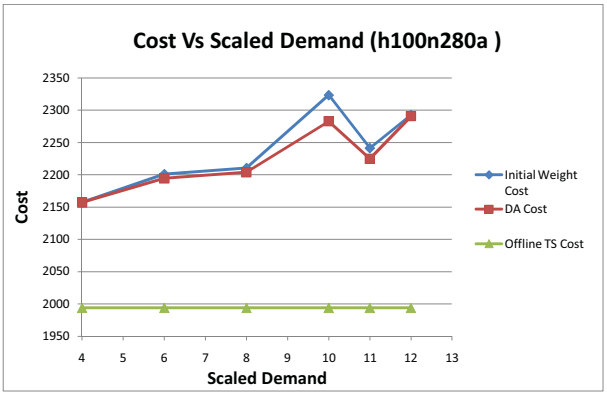

Figure B.4: Cost Versus scaled demand on h100n280a network.

B.1.5 h100n360a

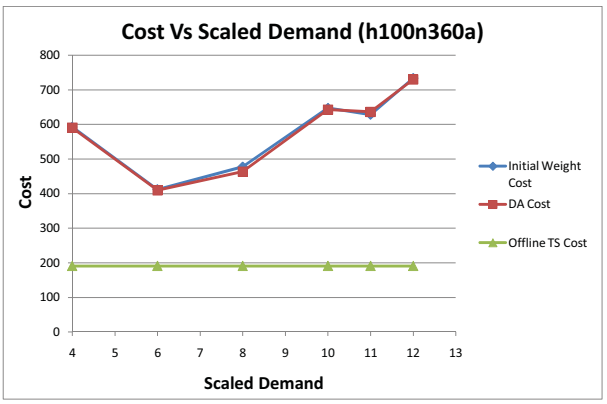

Figure B.5: Cost Versus scaled demand on h100n360a network.

B.1.6 r100n403a

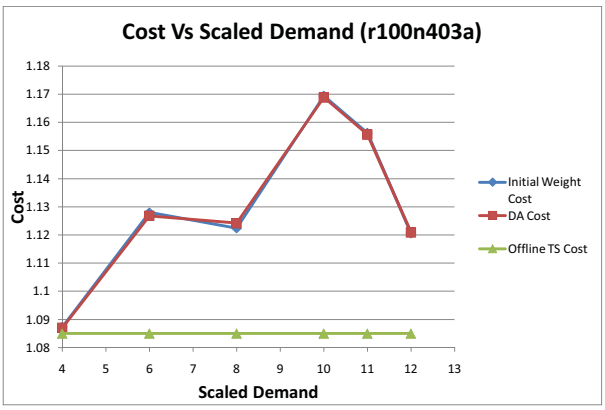

Figure B.6: Cost Versus scaled demand on r100n403a network.

B.1.7 r100n503a

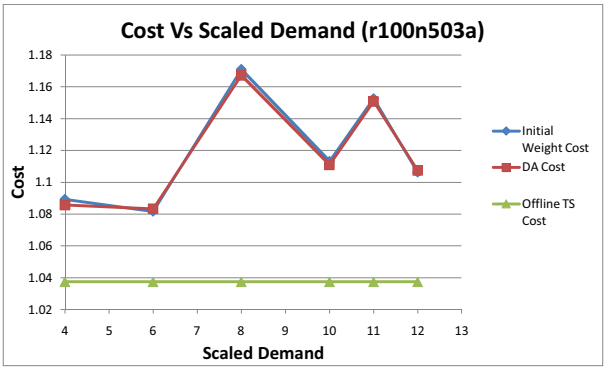

Figure B.7: Cost Versus scaled demand on r100n503a network.

Appendix-C

C.1 Validation

This section deals with the validation of the model and the code employed in the experimental results done and discussed in this thesis. The results that will be validated are of the Online Tabu Search code and the Deterministic Algorithm. To validate the results that have been generated by the Online Tabu search code and the Deterministic Algorithm, a simple method is proposed which is described in this section. The Online Tabu search results validation would be discussed followed by Deterministic Algorithm code.

C.2 Online Tabu Search

The Online Tabu search works with a random traffic demand and then starts looking for a better solution from the given initial solution. The initial solution used is one of the existing offline solutions. The Online Tabu search applies the initial solution weights to the given random traffic demand and then calculates the cost of these newly applied weights. After the cost is calculated, a search for better weights is done that would reduce the cost further. To verify the results, instead of searching for the weights for the random traffic demand, an existing standard traffic demand is taken, say for example D8 of r100n403a topology. The solution weights of D8 of r100n403a topology are given as initial solution. The process of validation is taking the solution weights of a particular traffic demand and are applied to the same traffic

demand and the cost calculated again. The cost of these traffic demand along with the solution weights are used from the earlier work done by Asad et al [25]. The cost calculated by the Online Tabu search for this traffic demand with the initial solution is compared to the existing results for the same work. If the results match that would imply the code is validated and is performing the process of applying the weights to the network under a given demand and calculating the cost correctly. Table C.1 shows the cost calculations done by the Online Tabu search algorithm along side the Offline Tabu search algorithm costs.

Table C.1: OLS Cost Validation Table for r100n403a.

Traffic Demand	**Online TS Cost**	**Offline TS Cost**
D4	1.0753	1.07528
D6	1.41093	1.41093
D8	2.04837	2.04837
D10	5.07246	5.07249
D11	15.0907	15.0907
D12	55.2688	55.2688

C.3 Deterministic Algorithm

To validate the Deterministic Algorithm code, an understanding of the process is needed. The DA works with applying the given initial solution weights to the demand at hand. Then the cost is calculated for applying the intial solution weights. Next, the DA does the demand processing and comes up with the new weights. The new weights are then applied and the new cost is calculated. The validation process

is done by applying the solution weights of a particular standard demand to the said particular demand and the cost is calculated. The cost, traffic demand, along with the solution weights are taken from the earlier work done by Asad et al [25]. So, in principle if we apply the same weight solution to the same traffic demand then the cost should match the cost achieved in the earlier work. To verify the costs that are calculated, a comparison is done with the earlier work results. To check the correct working of the code, instead of the random traffic demand DR, say D4 of h100n280a is used along with its weight solutions as the initial solution. If the cost calculated by DA by using D4 of h100n280a with its own solution weights as the initial solution matches with the cost of the work done earlier, that would mean proper working of the DA code cost calculation. Table C.2 shows the cost calculations done by the Online DA algorithm along side the offline Tabu search algorithm.

Table C.2: DA Cost Validation Table for h100n280a.

Traffic Demand	**Online DA Cost**	**Offline TS Cost**
D4	1.03836	1.03801
D6	1.15262	1.15207
D8	1.33599	1.33581
D10	3.30204	3.30254
D11	12.1186	12.1166
D12	20.1244	20.1210

C.4 Small Network Manual Application

This section describes the small network that is used for validating and understanding the DA. The network used is of 4 nodes with 10 arcs. Each arc has the capacity of 100. Two demand files are used, namely DX and DY. The Offline Tabu search is applied on the network with DX and the weights generated are saved as solution weights. Figure C.1 is the network used for applying the DA in a manual way.

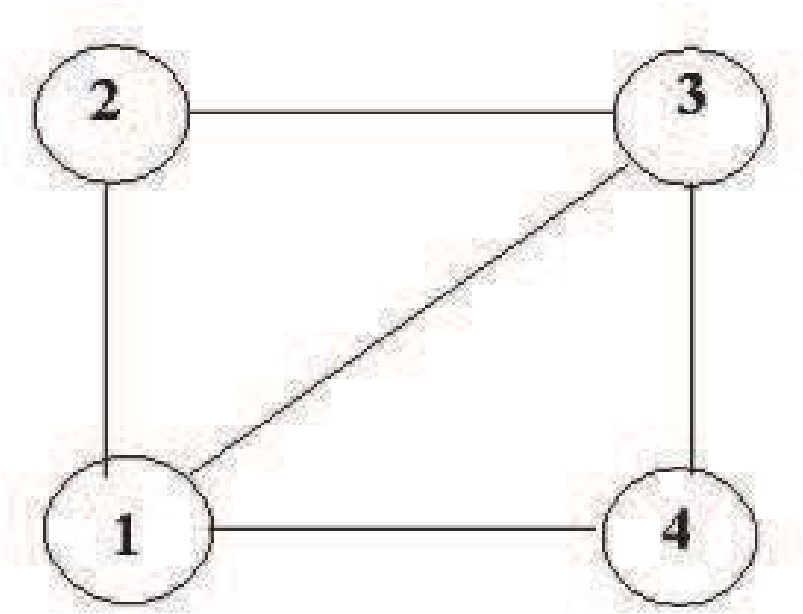

Figure C.1: Small Network of 4 nodes and 10 arcs.

The calculations that are done are as follows. First the DX and DY are processed so as to give the corresponding demand sum of the outgoing nodes. Table C.3 shows the results of the calculations done. In this table, column 1 shows the node numbers for which the column 2 and column 3 are calculated. In column 2 of Table C.3, the

sum of the demand of all the outgoing links from the nodes are computed. Column 3 of Table C.3 is the same as column 2 but for DY. Column 4 of the table has the weights computed using the Offline TS for the 4n10a network under traffic demand DX. The next column calculates the ratio of difference between the initial demand DX and the target demand DY. The last column weights is computed by selecting the highest demand ratio from the column DY/DX ratio and multiplying it with the corresponding weight from the initial solution to get the new weights.

Table C.3: DA weight Validation Table for 4n10a.

O/G nodes	**DX sum**	**DY sum**	**DX wts**	**DY/DX Ratio**	**New wts**
10	8.330724	3.842269956	5	0.461216811	5
9	1.900362	3.298288925	5	1.73561086	5
8	2.481323	4.688140084	8	1.889371148	8
7	4.009079	1.95879716	15	0.488590312	15
6	1.299134	2.767601141	6	2.130343091	12
5	8.522377	0.892069271	13	0.104673763	13
4	4.609075	1.752476044	8	0.380222939	8
3	4.307143	3.810820383	15	0.884767555	15
2	6.596945	3.647575684	5	0.552918917	5
1	3.193129	2.777029281	15	0.869689036	15

The cost of applying the demand DX weights to the demand DY of the 4n10a network is 0.0000152645. After the DA is applied and the new weights are found as written in the Table C.3, the cost is calculated again for these new weights. The cost of the new weights is 0.0000152640. There is only a single weight change from the initial solution. The calculations have been done by the program and were verified and validated manually. The results of the manual calculations are listed in Table C.3.

Vita

- Mohammed Moinuddin Rizwan Farooqi
- Received B.Tech (Bachelor of Technology) degree in Computer Science and Engineering from SHCST, Jawaharlal Nehru Technological University, Hyderabad, India in 2002.
- Joined Computer Engineering Department at KFUPM, Saudi Arabia in 2006.
- Completed M.S. (Master of Science) degree requirements in Computer Engineering at KFUPM, Saudi Arabia in 2010.
- **Nationality :** Indian.
- **Present and permanent address :** H. No: 8-2-156, Barkathpura, Nizamabad-503001, Andhra Pradesh, India.
- **Email address:** moinuddinrizwanfarooqi@gmail.com.
- **Mobile Number:** 00919848152168, 00966531835986.